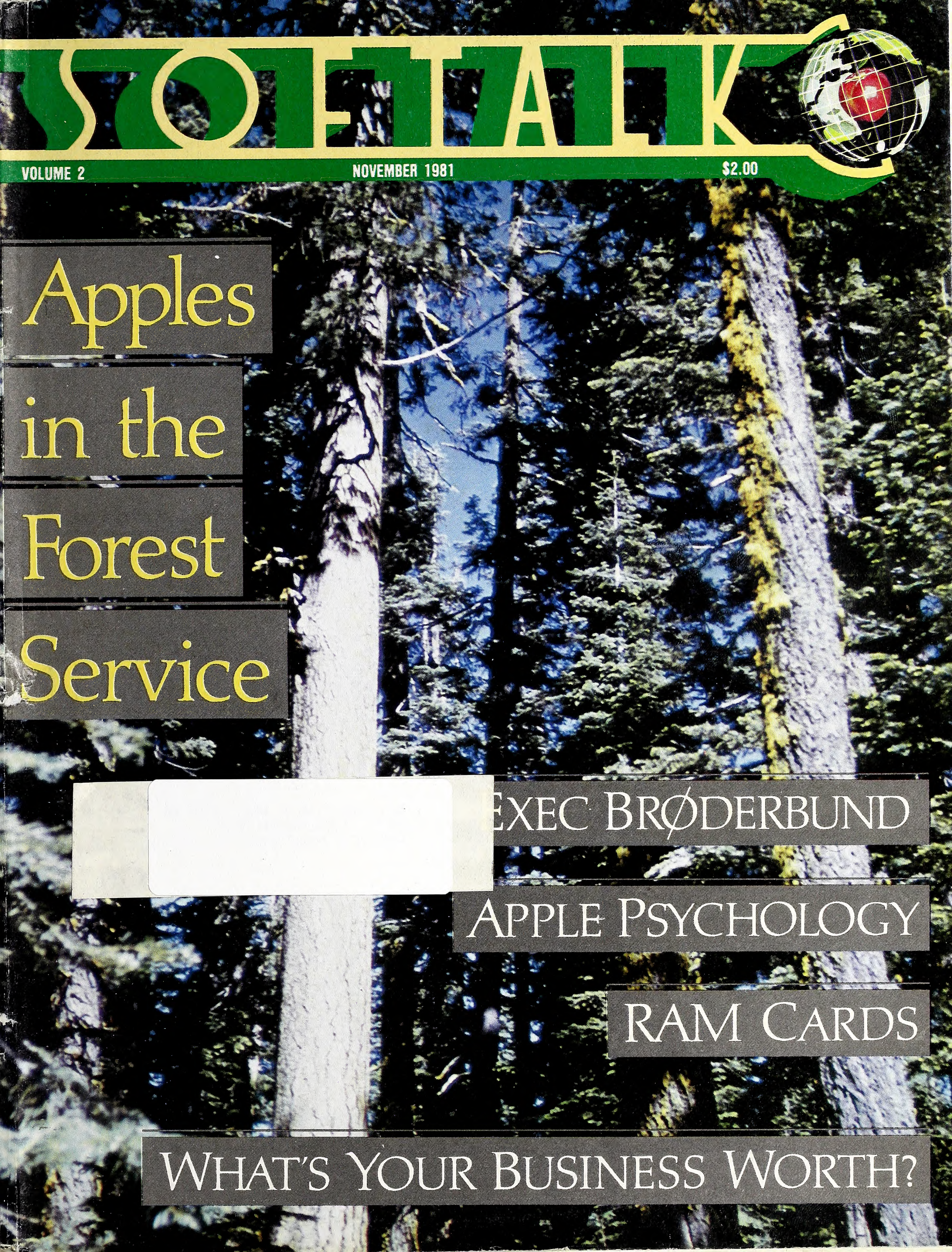
SOFTALK

VOLUME 2 NOVEMBER 1981 $2.00



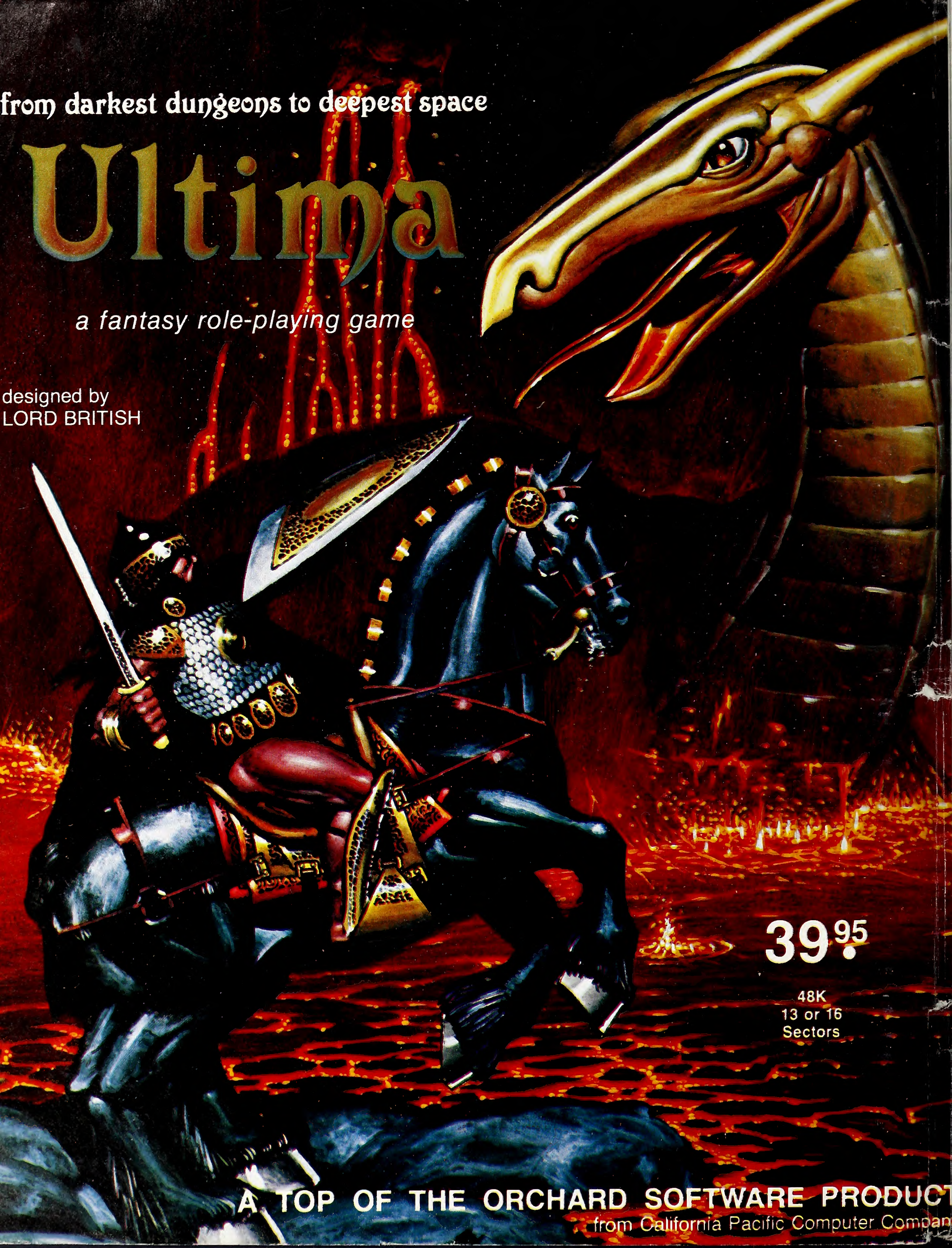
Apples in the Forest Service

EXEC BRØDERBUND

APPLE PSYCHOLOGY

RAM CARDS

WHAT'S YOUR BUSINESS WORTH?


from darkest dungeons to deepest space
Ultima
a fantasy role-playing game
designed by
LORD BRITISH
39.95
48K
13 or 16
Sectors
A TOP OF THE ORCHARD SOFTWARE PRODUCT
from California Pacific Computer Compan

SOFTALK

Chairman John Haller
Publisher Al Tommervik
Editor Margot Comstock Tommervik
Managing Editor Craig Stinson
Art Director Kurt A. Wahlner
General Manager Mary Sue Rennells
Editorial Associate Jean Varven
Assistant Editor Melissa Milich
Product News and Copy Editor Carol James
Contributing Editors
Assembly Language Roger Wagner
Pascal Jim Merritt
Business Peter Olivieri
Apple CP/M Greg Tibbetts
Apple III Taylor Pohlman
Hardware Jeffrey Mazur
Investing Kenneth Landis
Basic Solutions Wm. V. R. Smith
Special Projects Donna Siebert, Greg Voss
Ad Coordinator David Hunter
Circulation Kimberly Curling, Ron Rennells, Robert Mann
Art Assistant Miriam Gamble
Systems Wm. V. R. Smith, Ken Williams
Advertising Sales Al Tommervik
213-980-5074

Cover design by Kurt Wahlner; photo courtesy of United States Forest Service, Pasadena, California.

Composition by Photographics, Hollywood, California. Printing by Volkmuth Printers, Saint Cloud, Minnesota.

ISSN:0274-9629. *Softalk* is published monthly by Softalk Publishing Inc., 11021 Magnolia Boulevard, North Hollywood, CA 91601. Telephone, (213) 980-5074. Second-class postage paid at North Hollywood, CA, and additional mailing offices.

Postmaster: Send address changes to Softalk, 11021 Magnolia Boulevard, North Hollywood, CA 91601.

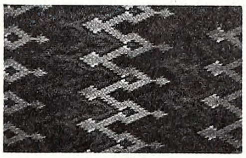
Subscriptions: Complimentary to all owners of Apple computers in the USA. If you own an Apple, but you're not receiving *Softalk*, send your name, address, and Apple serial number with a request for subscription to Softalk Circulation, 11021 Magnolia Boulevard, North Hollywood, CA 91601. *Softalk* is totally independent of Apple Computer Inc.; sending your warranty card to Apple Computer will not inform *Softalk* of your existence. Non-Apple owner subscriptions for one year: $12. For the convenience of business and schools in which several staff members share an Apple and would like individual copies of *Softalk*, multiple subscriptions to the same address are available at lower rates: two through five, $9 each; more than five, $7 each.

Back Issues: $2 through February 1981; $2.50 thereafter. November and December 1980 and January and March 1981 issues are sold out. *Softalk* will send you a back issue of your choice free (available issues only) for the name, address, and serial number of each Apple owner you can find who isn't already receiving *Softalk*.

Problems? If you haven't received your *Softalk* by the tenth of the month, or if you have other problems with your subscription, Ron Rennells or Kimberly Curling can help out. Call (213) 980-5099.

Moving? Send new address and old to Softalk Circulation, 11021 Magnolia Boulevard, North Hollywood, California 91601; telephone, (213) 980-5099.

CONTENTS



Page 60

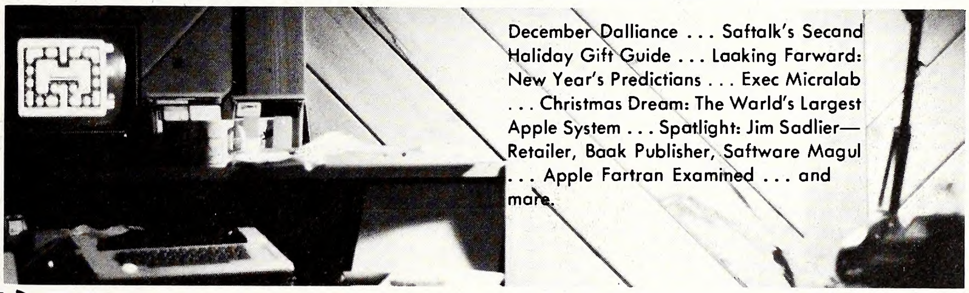
Page 68

Page 81

FEATURES

DEPARTMENTS

PREVIEWS



December Dalliance . . . Softalk's Second Holiday Gift Guide . . . Looking Forward: New Year's Predictions . . . Exec Microlab . . . Christmas Dream: The World's Largest Apple System . . . Spotlight: Jim Sadlier—Retailer, Book Publisher, Software Mogul . . . Apple Fortran Examined . . . and more.

ADVERTISERS INDEX

CONTEST: ART GALLERY

design a software package!

The wonderful thing about creating art on the Apple is that you don't have to be artistically talented to get a good result. You can just have fun and end up with a professional-looking work.

This month's contest is intended to pamper the artist in all of us.

The idea is to design a poster or package for your favorite program, real or imagined. The contest has two divisions. The first division is for entries designed on and printed from an Apple. The second division is for those of you who don't have access to a printer with graphics capability; entries for this division may be executed in any two-dimensional medium and need not be computer generated.

Six entries in each division will be selected by *Softalk*'s staff on the basis of originality, applicability of design to product, cleverness, and eye appeal. The twelve finalists will be printed in *Softalk*, at which time *Softalk*'s readers will have the opportunity to vote for their favorites to determine the two final winners.

The first-place winner in each division will receive $100 worth of goods produced by *Softalk*'s advertisers. The five runnersup in each division will receive $10 credits at their local computer stores.

All entries will be judged in black and white for the preliminary round. The twelve semifinal winners will be asked to submit printable-quality color photos or printouts of their works done on the computer or color versions of their hand-done art, which will be the versions printed in *Softalk*.

Send your entry with the coupon below to Softalk Art Gallery by December 15, 1981.

Name: ____________________

Address: ____________________

City/State/Zip: ____________________

____

Phone: ____________________

Dealer: ____________________

Prize You'd Like: ____________________

____

Your Autograph: ____________________

Mail this coupon with artwork to Softalk Art Gallery, 11021 Magnolia Boulevard, North Hollywood, CA 91601, by December 15, 1981.

U.S. Postal Service
Statement of Ownership and Circulation
(Required by 39 U.S.C. 3685)

1. Title of publication: Softalk
 A. Publication No. 02749629
2. Date of filing: October 1, 1981
3. Frequency of issue: monthly
 3A. No. of issues published annually: 12
 3B. Annual subscription price: Free to Apple computer owners

4. Location of known office of publication: 11021 Magnolia Blvd., North Hollywood, Los Angeles County, California 91601

5. Location of the headquarters or general business offices of the publishers: 11021 Magnolia Blvd., North Hollywood, Los Angeles County, California 91601

6. Names and complete addresses of publisher, editor, and managing editor:

Publisher: Al Tommervik, 11021 Magnolia Blvd., North Hollywood, CA 91601

Editor: Margot Comstock Tommervik, 11021 Magnolia Blvd., North Hollywood, CA 91601

Managing Editor: Craig Stinson, 11021 Magnolia Blvd., North Hollywood, CA 91601

7. Owner (if owned by a corporation, its name and address must be stated and also immediately thereunder the names and addresses of stockholders owning or holding 1 percent or more of total amount of stock. If not owned by a corporation, the names and addresses of the individual owners must be given. If owned by a partnership or other unincorporated firm, its name and address, as well as that of each individual must be given. If the publication is published by a nonprofit organization, its name and address must be stated.):

Softalk Publishing Inc., 11021 Magnolia Blvd., North Hollywood, CA 91601

Robert Comstock, 11021 Magnolia Blvd., North Hollywood, CA 91601

William Depew, 10432 Burbank Blvd., North Hollywood, CA 91601

John Haller, 11021 Magnolia Blvd., North Hollywood, CA 91601

John Mitchell, 10432 Burbank Blvd., North Hollywood, CA 91601

William V. R. Smith, 10432 Burbank Blvd., North Hollywood, CA 91601

Softape, 10432 Burbank Blvd., North Hollywood, CA 91601

Margot Comstock Tommervik and Al Tommervik, 11021 Magnolia Blvd., North Hollywood, CA 91601

Kurt A. Wahlner, 11021 Magnolia Blvd., North Hollywood, CA 91601

8. Known bondholders, mortgagees, and other security holders owning or holding 1 percent or more of total amount of bonds, mortgages or other securities (If there are none, so state) None

9. For completion by nonprofit organizations authorized to mail at special rates (Section 132.122, PSM) Not applicable

10. Extent and nature of circulation	Average no. copies each issue during preceding 12 months	Actual no. copies of single issue published nearest to filing date
A. Total no. copies printed (net press run)	43,283	47,500
B. Paid circulation		
1. Sales through dealers and carriers, street vendors and counter sales	4,443	6,400
2. Mail subscriptions	0	0
C. Total paid circulation (sum of 10B1 and 10B2)	4,443	6,400
D. Free distribution by mail, carrier or other means, samples, complimentary, and other free copies	38,000	40,559
E. Total distribution (Sum of C and D)	42,443	46,959
F. Copies not distributed		
1. Office use, left over, unaccounted, spoiled after printing	500	250
2. Returns from news agents	340	291
G. Total (Sum of E, F1 and 2 should equal net press run shown in A)	43,283	47,500

11. I certify that the statements made by me above are correct and complete.

Al Tommervik, Publisher

Can YOU Survive?

BUG attack

The bugs are on the loose, and they're hungry.
Defend against ants' antics, malicious millipedes, and the menacing medfly invasion.
Bug Attack will keep the whole family captive with colorful creatures and superb sound effects.
Expert de-buggers who defeat the bugs in the three gardens beware, as they must face a five-fly formation with a deadly flight pattern!

Defeat the Bug Attack at your local computer store.
Written by James L. Nitchals

Bug Attack requires Apple II, 48k. (13 or 16 sector).
*Apple is a Trademark of Apple Computer, Inc.

Brought to you by:

™ CAVALIER COMPUTER
P.O. BOX 2032 — DEL MAR, CA 92014 — (715) 755-8143

©1981 by Cavalier Computer

Exclusively distributed by

SOFTSEL
8295 So. La Cienega Blvd., Inglewood, CA 90301
(800) 421-5770 In Calif. (800) 262-4242

LittleCrypt Winner Found Delicious Answer on *McIntosh* Apple

SuperCrypt. Dann McCreary's Apple still belongs to Dann McCreary. As was expected all around, no one was able to break the SuperCrypt code; but three people did some excellent spadework as a start on it and one happy person actually took a flyer on a translation. Whether the latter is a "reasonable" start or not is a moot point—but we believe it is because it matches the spirit and, well, it could have been close.

E. N. Hendrick of Torrance, CA, whose letter—which was dated 11:21 P.M., September 15, or 39 minutes before deadline—described some remarkable, deep, clear cryptological thinking, begged an extension. Brian Holland of Endwell, NY, didn't attempt a translation but explained the methodology he believed would have to have been used to derive the code. R. F. Lauer of Nutley, NJ, went at the problem like a detective and came closest to understanding how the process really works. Because Lauer's entry included a letter with thoughts of more general interest, you can find it in Open Discussion. Finally, Sean Walter of Kansas City, MO, is the reader who took a chance on a real translation, which was:

"Hello! This is SuperCrypt. If you stop at the end of the next sentence, you automatically win. You may continue to find out if the third line really says what it should, but you automatically win if you stop at this period."

All four people will receive Absolute Security disks for their honest attempts.

Dann McCreary's translation of the SuperCrypt code remained sealed in an envelope under lock and key at Softalk throughout the contest period. In the last week of September, we opened the envelope, and here is the translation we found:

> As I compose this challenge, images of mysteries, things hidden, secret words, and obscure sayings run through my mind. . . .
>
> Solomon: "It is the glory of God to conceal a matter; to search out a matter is the glory of kings. . . . A word aptly spoken is like apples of gold in settings of silver." Proverbs 25:2.11
>
> Plato, Allegory of the Cave: "When any of them is liberated and compelled suddenly to . . . look towards the light, he will suffer sharp pains; the glare will distress him, and he will be unable to see the realities of which in his former state he has seen the shadows;"
>
> The Prophet Isaiah: "The Lord . . . who overthrows the learning of the wise and turns it into nonsense." Isaiah 44:25
>
> Distilled from *Covert Action Info. Bulletin*, Dec '80: The NSA intercepts "virtually all" phone calls, telegrams and telexes where one or both parties is outside the U.S., as well as many such communications within the U.S.
>
> John the Apostle: "In the beginning was the word, and the word was with God, and the word was God." John 1:1
>
> Poe, *The Gold Bug:* "It may well be doubted whether human ingenuity can construct an enigma of the kind which ingenuity may not, by proper application, resolve."
>
> David Kahn, *The Codebreakers*: "So the answers again evade the cryptanalyst, formless, endless, the random one-time tape vanquishes him by dissolving in chaos on the one hand and infinity on the other. Here indeed the cryptanalyst gropes through caverns measureless to man."
>
> Jesus: "For there is nothing hidden that will not be disclosed, and nothing concealed that will not be known or brought out into the open." Luke 8:17
>
> "I am the first and the last . . . and I hold the keys. . . ." Revelation 1:17,18

Victoria Wollner, winner of the Logical Choice announced in September, chose Gorgon as one of her prizes. Wollner resides in Anaheim, California.

LittleCrypt. The LittleCrypt paragraph said: "If James Bond, Mata Hari, and Napoleon Solo had used Apple computers, they would certainly have been called Northern Spies." Filling into the diagram the words if, had, Mata, would, James, called, Northern, computers filled the colored boxes with the letters I, H, T, O, M, C, N, S, which comprise an anagram of the word McIntosh, the name of a variety of apple. Besides the obvious connection—Apple itself—the cryptogram gave another clue. As the winner explained it: "Northern spy is an apple; as is McIntosh. Having the word Apple embedded in the code made the solution much easier." Do you recognize your words, Paul Rabson of Pittsford, NY?

Paul Rabson wins $100 worth of goods, which he'll pick up at his dealer, the Computer Center in Rochester, NY. Rabson broke the code for the paragraph in the easy way: the entire sentence code was created by touch-typing the real words with hands one row above and slightly to the left of appropriate position. Several entrants wrote programs to solve the cryptogram.

Because many entrants derived other real words (and some not so real) by entering different words in the boxes, we decided to give a $25 consolation prize to one of these entries.

The random number generator chose C. William Winkler of Canoga Park, CA, whose winning entry was the word *hypnotic*. Winkler will take his prize from Rainbow Computing in Northridge, CA. Other words submitted that can be made legitimately from filling in the boxes differently are acrimony, avionics, canopies, captains, captions, picayune, tympanic, and unchains. None of these, however, bears any relationship to the rest of the puzzle.

It's interesting that a large majority of entries that chose these words were multiple entries giving different words. Of the fifteen people who chose McIntosh, only one entered more than one answer, and that one disclaimed his action by writing that McIntosh was the answer, but that several other words would also fit.

OPEN DISCUSSION

Cal Pacific's Computer Camper Reports In

Thank you and California Pacific for sending me to Computer Camp. I had a great time and learned a lot. I entered a contest on the TI99/4, which I had never used before, and won first prize, which was a nylon jacket. I would have liked it to be an Apple jacket but I was happy to win it anyway.

I really enjoyed the camp and took all the classes I could. I took accelerated Basic, graphics, robotics, artificial intelligence, and game simulation. I met some real nice kids and had lots of fun. It was something I never dreamed I would get to do. I couldn't believe I was really going to the camp until I actually got there.

I applied to the director to work as a PA next year, so I hope I make it. If I don't, I am really happy to have been able to go once anyway. My family could never have afforded to send me and it would have taken me years to save up to go if I could.

The only thing I didn't like was that they didn't have enough Apples. Some of the kids in my cabin brought their own, so that helped. Maybe you can influence them to get a few more next time and also talk Apple into making them more available.

Anyway, thanks again. I really was happy to go.

John Brandstetter, South San Gabriel, CA

Believing in Security Can Be Perilous

Being an amateur cryptographer and a member of the American Cryptogram Association, I'd like to comment on Mr. McCreary's challenge cipher. First let me state that I have no desire to win his computer system, but I am concerned about his claims.

Since I am not a regular reader of your magazine (I own a Commodore PET), this cipher was brought to my attention by a stranger who read my letter to the editor in the July 1981 issue of *Creative Computing*. There I stated that it was dangerous for anyone not familiar with advanced cryptanalytic techniques to propose an enciphering scheme which they consider "absolutely secure."

In both the preamble to the cipher and in his press releases he makes exaggerated claims of security, which are not proved even if the challenge problem is not solved. This is dangerous since sensitive data might be enciphered, creating a false sense of security, while in reality it may be very vulnerable.

As is generally known among cryptographers, security cannot rest in the secrecy of the *general* system, but in the inability to recover the *specific* keys. I therefore *challenge* Mr. McCreary to publish his complete enciphering algorithm, less specific keys, if he has such faith in the invulnerability of his scheme. Lest anyone believe this to be unfair, remember that a determined cryptanalyst with access to an Apple can get a copy of the program, and even if a listing can't be gotten, then by proper inputting to the program, the actual details of the algorithm can be slowly determined in a short time. An analogy is an inventor of a new "pick-proof" lock who won't disclose its internal workings. Again, only a careful analysis can verify such claims.

The statement in his flyer, that it is "theoretically impossible to decode without the exact duplicate key" implies that perhaps a "one-time-pad" system is used. This involves using an infinite list of perfectly random numbers as the keying sequence, which is used only once. The production of the keying sequence is important, since *truly* random numbers are hard to compile and random number generators are easily deciphered from a relatively short segment.

In addition, the key sequence must be longer than all intended messages and must be produced, transmitted, and stored under high security conditions; loss of the key compromises all information, since it can't be retrieved. In cases such as these, one might as well store the raw information and avoid the costly encipherment. Such a system was abandoned long ago by the military as being impractical under normal working conditions and has also been ruled out by all computer-cryptographers for similar reasons.

Lest anyone assume that I'm out to "get" Mr. McCreary, let me say that I'm not. If his scheme is as good as he claims then he should certainly get all the rewards; if it is not as good, then such software should see very limited use, since a false sense of security is more dangerous than no security at all. Only by an unbiased evaluation of the enciphering algorithm can such a judgment be accurately made.

Rudolph F. Lauer, Nutley, NJ

Tops despite Omission

I read with great interest Craig Stinson's survey of the various home finance packages available for the Apple computer. I am very well acquainted with most of the products he mentioned in both his articles in April and May. Although, I must confess that I was very much disappointed when his review did not cover the Apple *Personal Finance Manager*.

As I stated, I have personally tried at least 50 percent of the products surveyed, but none has shown the ease of operation I have found with the *Personal Finance Manager*.

Like most of the products surveyed,

GOTO 15

Software Sale Sets Big Bucks Record

A new level of maturation of the microcomputer software industry into big business was achieved in mid November when Personal Software acquired for cash the rights to the source code of *VisiPlot* and *VisiTrend*, the best-selling business graphing packages from Micro Finance Systems.

No actual figures on the sale were released, but the transaction was known to be in the seven-figure range, making it the largest cash transaction in microcomputer history and the largest buy of any kind of rights to source code. *VisiPlot* and *VisiTrend* had previously been marketed by Personal Software under exclusive marketing agreements.

Other large sales of microcomputer software have either involved only finished product or have involved the entire publishing company. When Peachtree bought *Magic Wand*, it also assumed the ongoing business of the publishing company. Apple Computer has made several six-figure buys of finished product that do not include any rights to source code.

In making the announcement, Mitch Kapor, president of Micro Finance Systems and author of the two software packages, indicated that most of the proceeds from the sale would be pumped back into Micro Finance to fuel research and development of new software products.

PFM utilizes single-stroke entry as much as possible. It takes into account all cash, credit card, and check expenses. The program allows for establishing twenty-four budget codes to account for these expenses. (This can be somewhat limiting, depending upon the size of the family budget.)

The report formats allow for numerous reports ranging from monthly to yearly comparisons, complete with graphs, and yearly totals. The program allows for up to two hundred entries per month, a sum which I have found to far exceed my family's need—by approximately 75 percent.

One feature the *PFM* allows that I have not seen with other programs is a feature that allows a thirteenth month carry-over into the new year.

As I stated, I think the features of this program should be brought to the attention of Apple owners.

One last note, I really enjoy your columns on *VisiCalc* and assembly language.
Elton R. Shorter, Columbus, OH

Down in the Dumps

I recently purchased an Epson MX-80. I am very pleased with the printer but found the instruction manual left something to be desired as I am an Apple owner. Can you believe it contains a screen dump program for the TRS-80 but not for the Apple? No offense meant for TRS-80 owners.

As an Edu-Ware representative in my spare time (is there such a thing?), I get to travel most of the state of Florida visiting computer stores (a dream come true). In all my travels, I was unable to find a simple screen dump program.

Out of necessity, I decided to sit down one evening and write my own. Since necessity is the mother of invention—Eureka!—after four short hours, I had it. Seemed more like thirty minutes! I have enclosed a copy for your diligent perusal.

You may feel free to publish it for the benefit of all those poor souls out there that just bought an Epson (or other printer) and want to be able to dump their Apple screen using this carefully thought-out subroutine. It can be appended to any Basic program and called with a control-P.

A brief description of how it works may be in order. Line 33000 begins the subroutine and sends the action to line 33020 where the printer interface is told to wake up and get ready and the screen scroll is turned off. Lines 33030 to 33050 then take over the identify screen line locations and put them in the correct order for the printer.

Line 33010 is a GOSUB in the sub. This line identifies individual locations across each line on the screen and peeks them for the printer (neat, huh?). The return at the end of the line sends us back to the next I, which then sends us back through

OPEN DISCUSSION

from page 6

the GOSUB 33010 as many times as we need to in order to cover all the lines on the screen (eight times for each I).

After all twenty-four lines and forty screen locations in those lines have been peeked to the printer, we go to line 33060, which tells the printer interface thank you and good-bye and turns the screen back on.

It works well for me and my MX-80, and I don't know why it wouldn't work for any printer hooked to an Apple with little or no modifying.

```
10  REM **************************
15  REM *                        *
20  REM *  SCREEN DUMP PROGRAM   *
25  REM *                        *
30  REM *          BY            *
35  REM *                        *
40  REM *    ART CHRISTOPHER     *
45  REM *                        *
50  REM *        7/15/81         *
55  REM *                        *
60  REM **************************
65  REM
33000 GOTO 33020
33010 FOR J = I TO I + 39: A = PEEK(J): PRINT
      CHR$(A);: NEXT J: PRINT CHR$(13);: RETURN
33020 PR#1: PRINT CHR$ (9) "80N"
33030 FOR I = 1024 TO 1920 STEP 128: GOSUB
      33010: NEXT I
33040 FOR I = 1064 TO 1960 STEP 128: GOSUB
      33010: NEXT I
33050 FOR I = 1104 TO 2000 STEP 128: GOSUB
      33010: NEXT I
33060 PR#0
33070 REM
33080 REM ART CHRISTOPHER
```

Art Christopher, Boca Raton, FL

Fractious Fractions

I was pleased to read Richard Coleman's article on the use of the Apple II by the College of Home Economics at Louisiana Tech (August 1981). The "Fractions Maker" subroutine accompanying the article, however, violates several principles of good programming. In the interest of providing good examples of programming techniques, I think the following remarks are pertinent.

Although the program will run as published if input is made strictly as requested, an incorrect answer will result if a character other than "+" is used to separate the integer from the fraction; a space, for example, which is a natural way of typing a fraction. Or, try entering an integer followed only by a +. Beep! It appears that line 10060 was intended to check for this, but, since (II − 1) can never equal LEN(N$) because of the loop limit in line 10020, the If statement of line 10050 must always be true and line 10070 can never be executed—it is dead code.

Now, how did such an error creep into a perfectly good piece of code, and why isn't it immediately apparent? The answer is, because the subroutine is poorly structured—difficult to read through and observe the program flow.

One technique that helps is to avoid multiple exits from a subroutine. In the Fractions Maker, notice that there are three lines that contain returns from this one subroutine, depending on the conditions found in the input string. That may not seem too confusing when there is only one subroutine, as here, but it makes for a mish-mash when you write several subroutines in a block, which is common practice. Better to structure your code so there is a single entry and a single exit for each subroutine. If your subroutine logic really requires multiple exits, you can Goto the return line number from any earlier points from which you want to exit. The program will work the same way, but it will be much clearer when you or someone else tries to read and understand the source code.

Another suggestion is to make it clear in the coding where loops begin and end. The best way, when possible, is to make For . . . Next loops very short. When they stretch out over many program lines, the program flow is more difficult for us humans to follow. This is especially true if you branch out of the loop via If statements.

I enclose an example of a subroutine from my recipe management system, *Computer Cuisine,* which performs the same translation as the Fractions Maker. Aside from Rem statements, the two routines are about the same length. The input string is (AQ$) and the output value is (F). The input format accepts either a space or a hyphen separating the integer from the fraction.

I also enclose a subroutine for fractional output. If decimal numbers are inconvenient to input, they are doubly so when read on a screen or printer as quantities of sugar, flour, or eggs. The assumption upon which this subroutine is based is that, in food recipes, the only fractions needed are halves, thirds, quarters, eighths, and perhaps fifths. Given a real number input (V), this subroutine simply separates out the fractional part if any (line 140) then iteratively assigns a string value to F$ and tests the fractional real value against the midpoints between successively higher fractions. The first time the test fails, program flow falls through to lines 144 and 145 where the integer part, if any, of the original value is put together with the fraction.


CROSS FIRE

The result is the nearest common fraction, formatted in string (F$).

Computer Cuisine also does range checking for the units of measure, expressing scaled quantities in appropriate units (2 1/3 tablespoons rather than 7 teaspoons, for example). It also rounds certain units to whole numbers, as appropriate, so you are not told to use 1 7/8 cans of tuna.

```
10  REM ************************
11  REM *                      *
12  REM * FRACTIONS INPUT AND  *
13  REM *  OUTPUT SUBROUTINES  *
14  REM *        DEMO          *
15  REM *   DON RAVEY 8/1/81   *
16  REM *                      *
17  REM ************************
18  REM
20  TEXT : HOME
99  GOTO 199: REM ::SUBROUTINES::
100  REM OUTPUT FRACTIONS SUBR:
101 VI = INT (V):F = V - VI:F$ = " "
102  IF F > .0625 THEN F$ = "1/8": IF F > .1625
     THEN F$ = "1/5": IF F > .225 THEN F$ =
     "1/4": IF F > .291666667 THEN F$ = "1/3": IF
     F > .354166667 THEN F$ = "3/8": IF F >
     .3875 THEN F$ = "2/5": IF F > .45 THEN F$ =
     "1/2": GOTO 104
103  GOTO 105
104  IF F > .55 THEN F$ = "3/5": IF F > .6125 THEN
     F$ = "5/8": IF F > .645833333 THEN F$ =
     "2/3": IF F > .708333333 THEN F$ = "3/4": IF
     F > .775 THEN F$ ="4/5": IF F > .8375 THEN
     F$ = "7/8": IF F > .9375 THEN F$ = "":VI = VI
     + 1
105 VI$ = STR$ (VI): IF VI = 0 THEN VI$ = " "
106 F$ = VI$ + " " + F$: RETURN
108  REM
110  REM INPUT QTY SUBR:
111  FOR I = 1 TO LEN (AQ$):T$ = MID$ (AQ$,I,1)
112  IF T$ = "." GOTO 115: REM DECIMAL
113  IF T$ = "/" OR T$ = " " OR T$ = "-" GOTO
     116
114  NEXT
115 F = VAL (AQ$): GOTO 124: REM DECIMAL OR
     WHOLE NO., NO FRACTION
116  IF T$ = "/" THEN FW = 0:I = 0: GOTO 118:
     REM NO WHOLE NO.
117 FW = VAL ( LEFT$ (AQ$,I - 1)
118  FOR J = 1 TO LEN (AQ$): IF MID$ (AQ$,J,1) =
     "/" GOTO 120
119  NEXT
120 NF$ = "":FD$ = ""
121  FOR K = I + 1 TO J:NF$ =NF$ + MID$
     (AQ$,K,1): NEXT : REM NUMERATOR
122  FOR K = J + 1 TO LEN (AQ$):FD$ = FD$ +
     MID$ (AQ$,K,1): NEXT : REM DENOMINATOR
123 FF = VAL (NF$) / VAL (FD$):F = FW + FF
124  RETURN
199  REM ::::::MAIN PROGRAM::::::
500  PRINT "THIS IS A DEMO PROGRAM FOR
     FRACTION": PRINT "INPUT AND OUTPUT
     SUBROUTINES."
510  PRINT : PRINT "DO YOU WANT TO TRY THE
     INPUT OR OUTPUT"
520  POKE 34,5: REM SET TOP OF TEXT WINDOW
530  VTAB 5: CALL - 868: PRINT "ROUTINES, OR
     EXIT (I, O, OR X)? ";: GET A$: PRINT A$: HOME
540  IF A$ < > "I" AND A$ < > "O" AND A$ < >
     "X" THEN 530
550  IF A$ = "I" THEN PRINT : PRINT "INPUT A
     NUMBER ... EITHER DECIMAL OR": INPUT
     "FRACTIONAL    <LIKE: 15 2/3 > ";AQ$:
     GOSUB 110: PRINT : PRINT "THAT'S ";F: GOTO
     530
560  IF A$ = "O" THEN PRINT : PRINT "INPUT A
     DECIMAL NUMBER...": INPUT "<LIKE: 27.2865
     > ";V: GOSUB 100: PRINT : PRINT "THAT'S
     ";F$: GOTO 530
570  TEXT : END
```

Don Ravey, San Mateo, CA

Mixing Apples and . . .

I enjoyed Jeff Mazur's article "Terminal Data."

I truly hope you can shed some light on my problem. I live in New York City and have an Apple II with RAMCard, two disk drives, and Hayes Micromodem. My son lives in Binghamton where he bought a Radio Shack Videx terminal. This terminal produces thirty-two characters. It has a built-in modem. The Apple produces forty characters.

He succeeded in accessing the Apple and booted a disk. However, his readout on the TV screen was a wraparound on the same line. Undoubtedly, the two systems are not compatible.

Are there any software programs to make them work? Certainly would appreciate your consideration in this matter.

Irwin Horowitz, Bayside, NY

Direct Clutter

[To Greg Tibbetts:] Your new column in *Softalk* is long overdue and excellent in every respect. Hope you will continue to enlighten us SoftCard CP/M users and expand to other topics such as Microsoft Fortran and the Z80 assembler development package.

I am having an annoying occurrence with my CP/M, which may have already been corrected in the new version, which I have not received. Occasionally, especially when using CONFIGIO, I am getting files in the directory that I am unable to delete. While of no concern as far as storage is concerned, nevertheless, it clutters the directory. Any suggestions?

Would also appreciate it if you could cover in your column in detail how to enter more than a single assembly program to be used concurrently at the same hex address.

Henry Lustig, Vergennes, VT

A Policy Worth Applauding

In Allan Tommervik's article on Hayes Microcomputer Products (September 1981 issue), he mentions that Hayes has ". . . the most liberal warranty program in the Apple marketplace—two years." You might be interested in . . . our three-year warranty for our MC1 and MC16 Music Cards for the Apple. Usually, software is not covered by warranties, since a program bug may not be considered a manufacturing defect. To eliminate any doubt, our one-year warranty on software guarantees the software will work as described in the owner's manual.

I know that ALF has very high goals for product quality, and as a Hayes Micromodem owner, I applaud Hayes's quality goals and long-term warranty. I hope more companies will begin to recognize the importance of quality and start offering longer warranties than the usual

ninety-day ones, especially on software. After all, would you buy a system from someone who says, "Put your accounts receivable in the capable hands of our computer (software sold as is)"?
Philip Tubb, ALF Products, Denver, CO

Softalk, the Matchmaker

I am writing in reference to an article you published a year ago concerning Apples and the handicapped. At the time, you wrote, "There seems no appropriate way to end this story." I thought you might be interested in its continuation, which has come to involve me.

The emergence of personal computers has caught several people unprepared for this new technology. They are scared of computers, puzzled, uneasy. People in the computer field are still often viewed as unfeeling robots working with devilish machines bent on lousing up that next phone bill or magazine subscription. Friends of mine still insist that computers "will never catch on." I myself used to hate computers.

The story of how I became involved with the Apple began with that article in your magazine, and it's another human interest story that might show people that the world of computers is not robotic, insensitive, or dull. In fact, it can be the backdrop for romance.

In October 1980, *Softalk* published an article titled "Apples and the Handicapped." One of the persons featured in this article was David McFarling of Lincoln, Nebraska, a quadraplegic who owns Small Business Computer Systems, an accounting software firm.

At the time the article appeared, I had just been hired by Dave to write promotional material and to give nonprogrammer advice on his software. I took the job because I was unemployed and nearly penniless. For a while, there were a lot of doubts as to whether things would work out, and I began thinking of looking for another job. Then I read your article.

The description of his computer system went right over my head (what's a serial interface?), but Dave's stubborn determination to conquer his handicap impressed me. His spirit was captured by the author, and one of the quotes [quoting McFarling] stuck in my mind:

"Nothing, absolutely nothing, is impossible. True, there are some things the world is not yet ready for—or not yet willing to accept. When the chips are down, the difference between the possible and the impossible is the true measure of a man's ability."

"Nothing is impossible." I could learn about computers and be successful at my new job. I stayed on, and what happened later went beyond my expectations.

I'm now handling the advertising in addition to being a technical writer. Not only do I now know the difference between serial and parallel interfaces, but I'm learning to program on the Apple II Plus. The business is growing; it's moved out of the living room into a new addition built onto the house. Dave is still working hard and is now into Pascal.

The best part of the story I've saved for last. Dave and I were married August 15. He was concerned at first that I wouldn't be interested in computers and would be a "computer widow." But when I started staying up until 2:00 a.m. working on the computer and taking trade journals to bed with me, he knew we had similar interests.

I'd like to thank *Softalk* for its role in the romance. Dave was too modest then to tell me his accomplishments. I had to read about them in a magazine. Your article impressed me and gave me the determination to stick with a job I was unsure of, and I'm so glad I did.

The other night, I was working with one Apple, Dave on the other. Pausing for a moment, I asked, "Is this what our marriage is going to be like?" He replied that we could always send messages to each other via computer. But the message was already in his eyes: we may have a few bugs to work out, but this is one marriage programmed for happiness.
Diane Walkowiak, Lincoln, NE

Seeking New Beginnings

In response to a letter about Beginners' Corner, in which Tom Burt wrote in and said it was harmful to open up the disk drive door when the "in use" light is on: Well, the only real time it is dangerous to open the drive is when a write is in progress.

During a read, the computer just picks up information. If you typed in for the computer to load a program and you realized that you have the wrong disk in the drive, it would not hurt to open the door and change disks in the middle of operation.

I would have to agree with Tom that Beginners' Corner does not give much information. If it told us secrets about the Apple, like how to get other colors, short programs to help make shapes, how to make INT functions available in FP and vice versa, it would be better.
Chris Rys, La Jolla, CA

Scrambled or Sunny-Side-Up?

In view of the space-age program that is conquering the computer owners of this country, *Space Eggs,* I would like to give this brief point.

The biggest disappointment in playing the game is not, ridiculous as it is, the last fuzz ball sliding across the bottom of your screen to wipe out all hope of beating your high score, or being destroyed by a meandering spider in the screen following the successful navigation of the fuzz-ball field. The biggest disappointment, in my humble opinion, is finally redocking the first and second stages of your rocket, only to be blown up in the

QUALITY DISK SOFTWARE

BACKED BY ON-GOING APPLICATIONS SUPPORT

APPLE II Ⓐ TRS•80 Ⓣ

HOME FINANCE PAK I: Entire Series $49.95 Ⓐ Ⓣ

CHECK REGISTER AND BUDGET: This comprehensive **CHECKING ACCOUNT MANAGEMENT SYSTEM** not only keeps complete records, it also gives you the analysis and control tools you need to actively manage your account. The system provides routines for **BUDGETING INCOME AND EXPENSE, AUTOMATIC CHECK SEARCH,** and **BANK STATEMENT RECONCILING. CRT** or printer reports are produced for **ACTUAL EXPENSE** vs **BUDGET, CHECK SEARCH DISPLAY RECONCILIATION REPORT** and **CHECK REGISTER DISPLAY** by month. Check entry is prompted by user-defined menus of standard purposes and recipient codes, speeding data entry and reducing disk storage and retrieval time. Six fields of data are stored for each check: amount, check no., date, purpose, recipient and **TAX DEDUCTIBLE REMINDER. CHECK SEARCH** routines allow searching on any of these data fields. Up to 100 checks/mo. storage $39.95

SAVINGS: Account management system for up to 20 separate Savings accounts. Organizes, files and displays deposits, withdrawals and interest earned for each account $14.95

CREDIT CARD: Get Control of your credit cards with this program. Organizes, stores and displays purchases, payments and service charges for up to 20 separate cards or bank loans $14.95

UNIVERSAL COMPUTING MACHINE: $49.95 Ⓐ

A user programmable computing system structured around a 50 row x 50 column table. User defines row and column names and equations forming a unique computing machine. Table elements can be multiplied, divided, subtracted or added to any other element. Hundreds of unique computing machines can be defined, used, stored, and recalled, for later use. Excellent for sales forecasts, budgets, inventory lists, income statements, production planning, project cost estimates-in short for any planning, analysis or reporting problem that can by solved with a table.

COLOR CALENDAR: $29.95 Ⓐ

Got a busy calendar? Organize it with Color Calendar. Whether it's birthdays, appointments, business meetings or a regular office schedule, this program is the perfect way to schedule your activities.
The calendar display is a beautiful **HI-RES** color graphics calendar of the selected month with each scheduled day highlighted in color. Using the daily schedule, you can review any day of the month and schedule an event or activity in any one of 20 time slots from 8:00 A.M. to 5:30 P.M.

BUSINESS SOFTWARE: Entire Series $159.95 Ⓐ Ⓣ

MICROACCOUNTANT: The ideal accounting system for small businesses. Based on classic T-accounts and double-entry booking, this efficient program provides a journal for recording posting and reviewing up to 1,000 transactions per month to any one of 300 accounts. The program produces **CRT** and printer reports covering:

TRANSACTION JOURNAL	**BALANCE SHEET**
ACCOUNT LEDGERS	**INCOME AND EXPENSE STATEMENT**

Includes a short primer on Financial Accounting. (48K) $49.95

UNIVERSAL BUSINESS MACHINE: This program is designed to **SIMPLIFY** and **SAVE TIME** for the serious businessman who must periodically Analyze, Plan and Estimate. The program was created using our Universal Computing Machine and it is programmed to provide the following planning and forecasting tools.

CASH FLOW ANALYSIS	**SALES FORECASTER**
PROFORMA PROFIT & LOSS	**SOURCE AND USE OF FUNDS**
PROFORMA BALANCE SHEET	**JOB COST ESTIMATOR**
REAL ESTATE INVESTMENT	**INVENTORY ANALYSIS**

Price, including a copy of the Universal Computing Machine $89.95

BUSINESS CHECK REGISTER AND BUDGET: Our Check Register and Budget programs expanded to include up to 50 budgetable items and up to 400 checks per month. Includes bank statement reconciling and automatic check search (48K) $49.95

ELECTRONICS SERIES VOL I & II: Entire Series $259.95

LOGIC SIMULATOR: SAVE TIME AND MONEY. Simulate your digital logic circuits before you build them. **CMOS, TTL,** or whatever, if it's digital logic, this program can handle it. The program is an interactive, menu driven, full-fledged logic simulator capable of simulating the bit-time response of a logic network to user-specified input patterns. It will handle up to 1000 gates, including **NANDS, NORS, INVERTERS, FLIP-FLOPS, SHIFT REGISTERS, COUNTERS** and user-defined **MACROS.** up to 40 user-defined random, or binary input patterns. Accepts network descriptions from keyboard or from **LOGIC DESIGNER** for simulation . . . . $159.95 Ⓐ Ⓣ

LOGIC DESIGNER: Interactive **HI-RES** graphics program for designing digital logic systems. Draw directly on the screen up to 10 different gate types, including **NAND, NOR, INVERTER, EX-OR, T-FLOP, JK-FLOP, D-FLOP, RS-FLOP, 4 BIT COUNTER** and **N-BIT SHIFT REGISTER.** User interconnects gates using line graphics commands. Network descriptions for **LOGIC SIMULATOR** generated simultaneously with the **CRT** diagram being drawn . . . . $159.95 Ⓐ

MANUAL AND DEMO DISK: Instruction Manual and demo disk illustrating capabilities of both program (s) . . . . $29.95 Ⓐ Ⓣ

ELECTRONIC SERIES VOL III & IV: Entire Series $259.95

CIRCUIT SIMULATOR: Tired of trial & error circuit design? Simulate & debug your designs before you build them! With **CIRCUIT SIMULATOR** you build a model of your circuit using **RESISTORS, CAPACITORS, INDUCTORS, TRANSISTORS, DIODES, VOLTAGE** and **CURRENT SOURCES** and simulate the waveform response to inputs such as **PULSES, SINUSOIDS, SAWTOOTHS,** etc... all fully programmable. The output is displayed as an **OSCILLOSCOPE-STYLE PLOT** of the selected waveforms (Apple only) or as a printed table of voltage vs time. Handles up to 200 notes and up to 20 sources. Requires 48 **RAM** . . . . $159.95 Ⓐ Ⓣ

CIRCUIT DESIGNER: Interactive **HI-RES** graphics program for designing electronic circuits. Draw directly on the screen up to 10 different component types, including those referenced above. Components interconnect list for **CIRCUIT SIMULATOR** generated automatically. Requires $159.95

MATHEMATICS SERIES: Entire Series $49.95

STATISTICAL ANALYSIS I: This menu driven program performs **LINEAR REGRESSION** analysis, determines the mean, standard deviation and plots the frequency distribution of user-supplied data sets. Printer, Disk, I/O routines $19.95

NUMERICAL ANALYSIS: HI-RES 2-Dimensional plot of any function. Automatic scaling. At your option, the program will plot the function, plot the **INTEGRAL,** plot the **DERIVATIVE,** determine the **ROOTS, MAXIMA, MINIMA, INTEGRAL VALUE** $19.95

MATRIX: A general purpose, menu driven program for determining the **INVERSE** and **DETERMINANT** of any matrix, as well as the **SOLUTION** to any set of **SIMULTANEOUS LINEAR EQUATIONS.** $19.95

3-D SURFACE PLOTTER: Explore the **ELEGANCE** and **BEAUTY** of **MATHEMATICS** by creating **HI-RES PLOTS** of 3-dimensional surfaces from any 3-variable equation. Disk save and recall routines for plots. Menu driven to vary surface parameters. Hidden line or transparent plotting $19.95

ACTION ADVENTURE GAMES: Entire Series $29.95 Ⓐ

RED BARON: Can you outfly the **RED BARON?** This fast action game simulates a machine-gun **DOGFIGHT** between your **WORLD WAR I BI-PLANE** and the baron's. You can **LOOP, DIVE, BANK** or **CLIMB**-and so can the **BARON.** In **HI-RES** graphics plus sound $14.95

BATTLE OF MIDWAY: You are in command of the **U.S.S. HORNETS' DIVE-BOMBER** squadron. Your targets are the Aircraft carriers, Akagi, Soryu and Kaga. You must fly your way through **ZEROS** and **AA FIRE** to make your **DIVE-BOMB** run. In **HI-RES** graphics plus sound $14.95

SUB ATTACK: It's April 1943. The enemy convoy is headed for the **CONTROL SEA.** Your sub, the **MORAY,** has just sighted the **CARRIERS** and **BATTLESHIPS'** Easy pickings. But watch out for the **DESTROYERS** - they're fast and deadly. In **HI-RES** graphics plus sound $14.95

FREE CATALOG-All programs are supplied on disk and run on Apple II w/Disk & Applesoft **ROM** Card & **TRS-80** Level II and require 32K **RAM** unless otherwise noted. Detailed instructions included. Orders shipped within 5 days. Card users include card number. Add $1.50 postage and handling with each order. California residents add 6½% sales tax. Foreign orders add $5.00 postage and handling.

SPECTRUM SOFTWARE
142 Carlow, P.O. Box 2084
Sunnyvale, CA 94087

FOR PHONE ORDERS: (408) 738-4387
DEALER INQUIRIES INVITED.

first few moments of the game thereafter.

That is my opinion. Nasir, wherever you are . . . that is one very mean and tricky game!
Don Howe, Scotia, NY

Adventurer's Lot Is a Savage One
The vast volume of software for the Apple II has created a problem; even with ads and reviews, it is hard for the prospective user to judge before he buys it whether a program will be enjoyable.

Softalk reviewed briefly Scott Adams's *Adventure #10: Savage Island, Part 1.* I tried to play it for almost two months but finally *had* to give up and return the program to the computer store. It *may* be labelled "for experienced adventurers," but even this kind of person would be driven crazy with frustration.

The trouble lies in the random factors that determine the occurrence of fatal incidents (bear eats the person, storm carries him out to sea). Every time I was about to overcome these two obstacles and find out finally what fired the cannon on the atoll, the sun set on me and "wait" never brought the dawn. I finally had to give up for good, even with explicit hints from other players.

If I was able to find the random factor generator and disable it by replacing the random timing with fixed values so I would have a fighting chance, I would have continued. But with the bear, the storm, and the sun setting so I could not proceed, I was never given that chance.

I advise anybody who has not purchased *Adventure #10* not to buy it but to save his money for something else. Anybody who *has* bought it should return it to the store. It's far too unreasonable and frustrating (it will not save a game while the hurricane is raging, in fact). I've gotten all but one point in the Microsoft *Adventure* but am unable to get anywhere in *Savage Island.* Unless it is tamed a bit by disabling the random generator, I am afraid it deserves a negative review.

I hope that this letter will prevent a lot of people who are considering buying it from wasting their money, time, sanity, and health on this gantlet program.
Paul R. Wilson, Bergenfield, NJ

One man's meat is another man's hurricane.

Magic Methods Revealed
I read a letter from a John W. Butler of Apalachin, NY, in September 1981 issue of *Softalk*. John was asking about the use of *Magic Window* and the Epson MX-80 printer.

Well, I have an Epson MX-80 printer that I am using with *Magic Window,* and I was having the same problem with the ESC control codes for the MX-80. I called Artsci, the vendor for *Magic Window,* on this, and this is the solution to the problem:

Press control-B, then press the ESC key; this is then followed by the capital letter of control you want.

For example, control-B, ESC, cap E will tell the MX-80 to go into emphasis mode.

This can also be used with other MX-80 controls; however, I have found out that sometimes you cannot put more than one control sequence on a line.

I hope this solves John's problem as well as those of other users of *Magic Window* and the Epson MX-80 printer.
Michael Gibson, Louisville, CO

I would like to respond to a request for assistance from John Butler in his letter.

Mr. Butler states he is having problems interfacing his *Magic Window* word processor with an Epson MX-80 printer. There is a "Technical Notes" document from Artsci covering this subject.

The sequence Mr. Butler mentions is the one controlling emphasized printing. This sequence requires an ESC code in the text followed by an upper case E (on) or F (off).

Emphasized printing works on a minimum of one line of text.

The problem is the ESC code. It can't be entered directly. The *Magic Window* uses the ESC key to control upper/lower case selection. But, by examining the ASCII code chart on page 7 of the *Apple II Reference Manual,* [you'll find that] the ESC key generates the hex code 9B. This value is what the printer expects. Back to the chart; a *control*-ESC also generates 9B. So the answer is to enter the ESC code into the text using the control code command (control-B) of *Magic Window* and the ESC key.

The sequence becomes:
Control-B, ESC, E—for on
Control-B, ESC, F—for off
The commas are for readability and must not be entered.

I hope this helps Mr. Butler. I have been very happy with my *Magic Window* and the support documentation provided by Artsci.
Roy Trahan, Fullerton, CA

[To John Butler:] The problem you described . . . is probably caused by one of two conditions. The first is that any control or escape characters entered must be in upper case. Since, however, you do not seem to be having problems with any control other than the emphasized print, it is more likely that your problem is in the limitation on the Epson emphasized print mode. Any escape-E found in a line will cause the entire line to print in emphasized mode. The second rule is that you must not use the escape-F command (which terminates the emphasized print mode) in the same line as the escape-E. If both escape-E and escape-F are found in the same line, the printer will ignore

the escape-E, [printing] the line in regular print.

I have been using the *Magic Window* and Epson MX-80 with my Apple II and have found them very compatible and satisfactory.

Good luck with your problem.
Martin Tiersky, Lincolnwood, IL

From Soup to Nuts—or, from Wagner to Robots
Excellent! The September issue reaches new heights; and all for free, too. It's hard to believe. *Softalk* is by far the best Apple magazine available today. A couple of comments:

1. I hope Wagner's Assembly Lines is printed in book form. It is clearly excellent.

2. I'm becoming confused about the new memory expansion boards now available, including the recent 64K board. I wonder if you would consider reviewing them in a future issue. Specifically, I would like to know how I can get the maximum amount of memory available to me for programming. I already have a language card and understand I might be able to use a second 16K board on which to locate DOS, but I don't quite get it. Can you explain?

3. The article on the new compilers was excellent; however, I still have one question: can an Integer ROM machine make use of these programs? Some of the ads say "Applesoft ROM or equivalent." Does this phrase mean Applesoft in the language card is okay?

4. I just received my Robot poster. Thank you. Unfortunately, it arrived a bit wrinkled, and one of the mailing tube covers was gone. I was really hoping to frame it. Would it be too much trouble to send me another? I'd be glad to pay expenses.
Michael Schuyler, Kingston, WA

1. Look for Roger Wagner's Everyone's Guide to Assembly Language *in book form around Thanksgiving. In it are the first year's columns, revised and expanded, and some new material.*

2. No sooner said than done. See page 84.

3. RAM Applesoft loaded into the language card or into a RAM card is indeed the equivalent of ROM Applesoft.

4. As we said in letters to all purchasers, if your poster arrived damaged, simply let us know, and you'll receive another. We added an outer box over the double tube packing as soon as we heard some posters hadn't made it through the mails unscathed.

A Letter to On-Line
Before I begin, I ask you to turn to page 68 of the September issue of *Softalk*. It is the ad at the bottom of the page that brings me to write this letter.

I had hoped the computer industry could remain immune to the current wave of sex in advertising. You have proven that I was, indeed, naive. I cannot understand why you feel you must sell your software—"for adults only" or not—by using pictures of naked women. You could at least have the decency to pose your models lower in the hot tub, so as not to turn *Softalk*—an otherwise fine publication—into cheesecake.

Softalk is read by many Apple owners, and, while I myself am single, there are many owners with children who read their issue of the magazine. Finding an ad like yours in the magazine must certainly put these parents off. It is bad enough that their children's television viewing hours are riddled with sexual themes and suggestions. Now a child can't even attempt to work with a computer—an essentially sexless tool—without being besieged with these unnecessary ads.

I, for one, hope you will take a close look at your advertising strategy for this product—which, in my opinion, is as useless and unnecessary as its ad—and think about what you are doing to the computer industry. I feel you are lowering it to the gutter level of all the other industries that have used sex in advertising for so many years.
Richard Gillett, Manhattan, KS

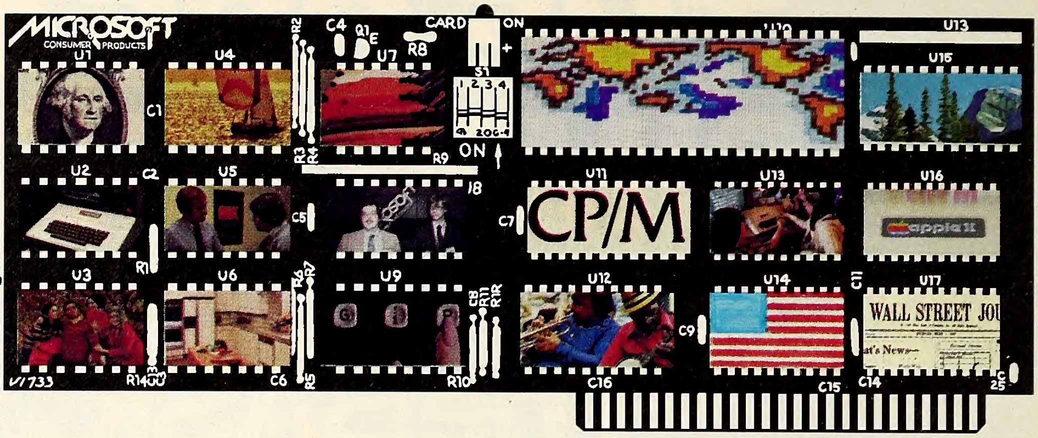
Vacuous Sex Symbol Misrepresents Women
Softalk has become a valuable resource in the classroom along with *Creative Computing* and others in my computer literacy class. Page 70 is not in good taste with your usual informative magazine. Women will never get out of the bedroom if this type of advertising is continued. We've come a long way, and it's up to us to complain when we are put down this way.
Mary Miller Smith, Doraville, GA

A Plea for Working Disks
Recently, I spent about $600 on Hayes Micromodem, *Visiterm*, and Source for my eighteen-month-old Apple. Although I've never had problems with my disk drives, neither software package would load, and it took several weeks and numerous trips to the local Apple store before I was up and running. It does not seem fair to me, to the store, or to Apple Inc. to market disks that are so marginally readable in the name of being copyproof. This was not cheap software, and so I have a right to expect quality!

Also, the most important single expectation of the software was that it switch incoming material to the printer (Silentype) when I want hard copy of names, addresses, and significant data. Some day I may want to transmit or receive files, but first of all I want a communications capability.
Arthur Radcliffe, Ann Arbor, MI

SOFTCARD Symposium

by Greg Tibbetts



Last month we left off our continuing series on Basic-80 after defining those of its commands that do not exist in either of the Apple Basics. This month we'll begin a discussion of Basic-80's disk file commands and the ways in which CP/M handles data files differently from Apple DOS.

Among other things, we'll examine in depth the logical and physical structures of text files, since an understanding of these matters will help you optimize your files for speed and disk space. We'll deal with the primitive storage of data on the disk only to the degree necessary to explain differences between CP/M and DOS. For those of you who might wish a closer examination of the more esoteric aspects of disk file handling, we may take up that topic in a future column.

We should point out that, while anything stored on a disk is in fact a disk file, we will be using that term interchangeably with data file and text file; all three of these terms may be taken to mean data files created from Basic.

Under both CP/M and DOS, there are two types of files: sequential access and random access; the names are usually shortened to sequential and random. As the words indicate, the general difference between the two types lies in the way information is accessed by the operating system when it reads from and writes to the file. This, in turn, is dependent on the way data is stored within the file.

The way the data is stored defines the file's format or structure. Before proceeding, we should define a couple of terms so we'll be on common ground.

The term *byte* refers to a unit of information manipulated and stored by the computer. In the case of data files, where more often than not data is stored in the form of ASCII characters, the byte is the smallest unit of the file and equivalent to one ASCII character. Therefore, the word *hello* in a text file would be made up of five bytes. The term *data element* means a logical unit of data defined by you, the programmer. It may be composed of one byte or many bytes of physical data, but it's always a complete, logical unit to you.

Stated simplistically, sequential files store data elements one byte after another, with the elements packed together and separated only by a delimiting character—usually a carriage return. No intervening spaces exist—except those you provide—so the file is a stream of bytes occurring in the exact order in which your program wrote them.

Random files, on the other hand, can be thought of as a number of boxes of identical size. These boxes are called *records*: their size is determined by you when you create the file and must be specified each time the file is accessed. Each record may contain one or a number of data elements, at your discretion, but the smallest unit of the file that Basic can recognize is the record.

The record supplies you with a fixed number of bytes of space, and what you do with them is totally up to you. The only thing Basic will do is add a carriage return at the end of your data in Apple DOS, or fill up any unused space in the record with the ASCII code for space in Basic-80.

At this point, the reason for the names sequential and random becomes obvious. Because all the data is in sequential files is packed together, Basic must effectively examine every byte in order until it finds the byte or bytes you want to access. On the other hand, since random files are made up of records of identical size, Basic knows—through calculation—how far into a file to go to reach a specific numbered record. It can, therefore, access any given record directly without having to read all the previous records.

Both types of files have their advantages and disadvantages. Random access files are generally longer, since they contain the wasted area that Basic padded with spaces; however, data access in a random file is generally faster, since Basic can go immediately to the record being sought. For the same reasons, sequential files, while more efficient users of space, are generally slower to access.

Because there are no inherent limits on lengths of data elements in sequential files, while there are in random, the two file types differ in the amount of planning they require of a programmer. In a sequential file, Basic needs no information concerning the length of the data element since it knows it has the entire element as soon as it encounters the delimiter. Therefore, in this kind of file, you can intermix different types and sizes of data with little advance planning.

The random file situation is different. Since the operating system has to be told by you, the programmer, how many bytes constitute a record, you will need to do some thinking about the maximum sizes of data elements and the overall structure of the file—before you write a single byte of data.

The fact that sequential files require less forethought, however, does not make them better than random files in all situations. In fact, many programmers are led by this apparent simplicity to expend unnecessary effort trying to get a sequential file to do an effective job on their data, when the type of data is really better suited to a random file structure.

Now that we're reasonably sure we're all talking about the same things when we talk about the types of data files, let's take a look at the similarities and differences in the ways text files are handled under Basic-80 and Apple's Basics. Apple text files, for all intents and purposes, have the same characteristics whether they are created by Applesoft or Integer Basic, so we will treat files created by both of these languages as features of Apple DOS.

In Apple DOS, sequential file manipulation is done in the traditional sense, as described above. All data is stored as ASCII representations of individual characters, with each data element separated by a single carriage return as a delimiter. In fact, the only difference between the format of Basic-80 text files and those made by Apple DOS is the fact that Basic-80 uses a carriage return and a line feed together as a delimiting pair. Although there are differences in command syntax—which we will discuss shortly—the logical structure of sequential files under both systems is the same: data element, delimiter(s), data element, delimiter(s), and so on. Samples of the actual disk images for the same sequence of data elements in DOS and CP/M files are shown in figure 1.

With regard to random access files, however, there are major differences in the logical structure itself. Before we attack these differences, let's take a look at the physical structure of both Apple DOS files and CP/M files. The terms we'll

EXAMPLE: For the three data elements APPLE, PEAR, and the number −27, the disk images appear as follows:

APPLE DOS:

A	P	P	L	E	[CR]	P	E	A	R	[CR]	−	2	7
C1	D0	D0	CC	C5	8D	D0	C5	C1	D2	8D	AD	B2	B7

BASIC−80

A	P	P	L	E	[CR]	[LF]	P	E	A	R	[CR]	[LF]	−	2	7
41	50	50	4C	45	0D	0A	50	45	41	52	0D	0A	2D	32	37

Figure 1.

NOTE: All numbers above are in hexadecimal, and the differences in the values shown are a result of the Apple's use of something called negative ASCII in which all values are exactly 128 decimal, (80 hex), higher than normal. See your Apple Reference Manual for a list of these codes.

need are: track, sector, block, extent, allocated, unallocated, track/sector list, and directory.

Track refers to a ringlike area on the disk where data is written by the disk drive. All Apple Disk II drives provide for thirty-five such circular areas, each slightly smaller than the one outside it. They are numbered 0 through 34, from outermost to innermost. Each track is further subdivided into a number of equal-size sections called sectors. On the Apple, the number of sectors per track is either thirteen or sixteen; our discussion is applicable to either case.

Obviously, the higher the track number and the smaller the ring comprising the track, the less physical distance is represented by a sector. In the Apple, however, whether Apple DOS or CP/M is used, each sector consists of exactly 256 bytes, with those sectors nearer the center compressing more bytes per unit of distance.

This 256-byte sector is the smallest logical and physical unit that can be manipulated by Apple DOS and is, in fact, the only unit it deals with. CP/M, however—because it needs to be portable between machines that may range from 128-byte sectors to 2,048-byte sectors—has adopted a 128-byte logical sector.

When CP/M is used on the Apple, the physical sector—the smallest unit the drive can handle—is still 256 bytes, but CP/M is kept ignorant of this fact by translations that take place in the BIOS. The important point to remember here is that, while Apple DOS recognizes only 256-byte structures, both logically and physically, CP/M deals with 128-byte logical sectors and 256-byte physical sectors.

The next larger logical unit that CP/M recognizes is known as a block. The term *group* is also used to describe this unit, but we'll use *block* exclusively, to avoid confusion. Since Apple DOS has no equivalent unit, CP/M is free to define the term without fear of conflict; hence, a block in Apple CP/M is always 1K bytes. CP/M numbers blocks from 0 to 131, starting at track 2, sectors 0 through 3, and ending at track 34, sectors 12 through 15.

The largest logical unit of disk storage used by CP/M is the *extent.* Files longer than 16K are divided into logical extents of sixteen blocks each. These extents are numbered from 0 through the amount needed to complete the file. The concept of sectors, blocks, and extents as three graduated units of disk file structure will be important as we proceed, so be sure you have a firm grasp of these terms.

Now that we have some units to deal with, we can talk about their allocation. Both CP/M and Apple DOS set aside certain space on the disk whenever a file is created or added to. This process is known as *file allocation.* It can be seen, then, that any disk in use is made up of allocated and unallocated space, and that both CP/M and Apple DOS must deal in certain minimum units of allocation. In DOS the unit of allocation is again the 256-byte sector, while in CP/M the minimum unit of allocation is the 1K block.

Obviously, both systems need a way to keep track of the allocated and unallocated space, and both do it by using special types of files called *directories.* Directories are simply a collection of records—35 bytes each for Apple DOS, and 32 bytes each for CP/M. The directories contain information such as the file name, status (active or deleted), some type of list (CP/M), or pointer to a list (DOS), of storage units allocated to the file. In DOS the list pointed to by the directory entry for the file is called a *track and sector list;* in CP/M the list is called a *block allocation table* and is contained in the last sixteen bytes of the directory entry.

By now some of this should be coming together for you. For example, in CP/M—since there are sixteen bytes available in the entry for listing block allocation—a maximum of sixteen blocks (16K of allocated space) can be listed. If more is required, another directory entry (the second extent) is created, and sixteen more blocks can be listed.

In DOS, the directory entry's track and sector pointer indicates a specific sector where there is a list of two byte entries (track and sector) that make up a complete map of the file's allocated space. Each sector can hold 122 such entries and, if necessary, a pointer to another sector where the track/sector list is continued.

But do we really need to know all this to understand random file storage? Possibly not, for a superficial understanding, but we implied at the beginning that confusion about what went on below the surface was the cause of a lot of inefficiency in random file usage; and one of the most confusing aspects of random files is how Basic is able to add records between other records, without having previously allocated the space.

Now it's fairly obvious that if Basic had access to an imaginary 8.5 million-byte disk, and could use this disk to store a single file of 32,767 256-byte records, then it could allocate physical space for the entire file. In that case, it would be easy for it to calculate the track and sector to read and write when you asked for a particular record from number 1 to number 32,767 (the range of random record numbers allowed by both systems). Our real disks, however, are of only 126,000 bytes total capacity, and although we cannot store all 32,767 records, we still require Basic to access any record number within that range.

You need just one more piece of the puzzle to see how that's done. The answer is that, while the system does not allocate

space for the actual records, it does allocate space in the block allocation table and in the track/sector list for all those entries that might eventually exist physically. For example, in CP/M, when Basic is told to create a file and write record 1, it creates the directory entry in block 1 or 2 (reserved for the directory only), finds an unused block elsewhere on the disk, writes the record, and then enters the block number in the block allocation table as the first entry. Later, when it's told to write record 32,767, it calculates the number of the extent where such a record would be found if it existed. It then looks for the extent and, if it cannot find it, creates it. Next, it finds another unused block, writes the record, and, finally, enters the block number in the block allocation table of that extent where it would appear if the extent were fully allocated. So storing record 32767, rather than costing 8 million bytes of allocated space, has cost an additional block and another thirty-two bytes in the directory.

Apple DOS performs similarly, except that it also creates all the empty track/sector lists in between, which—for the example given—would cost 262 sectors of additional allocated space. By now you should already be aware of how proper planning of things like record number ranges—rather than just taking the easy way out by using people's social security numbers, for example—can greatly increase your file storage efficiency.

Now we'll look at the specific differences in the handling of data elements and the general command differences between Apple DOS and Basic-80.

The only new terms we will need are *buffer* and *field.* The general definition for buffer is a space reserved in memory to store temporarily a series of bytes—either prior to, or following, an I/O operation. One example is the buffer used to store keyboard input on the Apple until you hit return.

With regard to random files, the buffer is a space in memory that stores the contents of the disk sector containing the record or part of a record you are working on. The smallest unit of data the Apple drive can move in or out of the buffer is the disk sector, so when you manipulate data elements in a single record, you are actually altering only the contents of the buffer. The disk file is not updated until you either close it, read or write another record, or—in the case of large records—begin manipulating a part of the record that appears in a subsequent sector.

The buffer itself is divided into sections called *fields*. They may be defined implicitly by the system as a single data element, as in Apple DOS, or they may be under the complete control of the programmer, as in CP/M. They are a logical unit, however, and vary in size; therefore, they should not be considered in the same way we consider fixed-size units like blocks and sectors. The concept of fields should become easier to understand as we discuss random files, but it's important to remember they're a logical construct that's meaningful only to the buffer, not the disk file itself.

While Apple DOS does do the primary job of allowing a programmer to access individual records randomly, it treats the record itself as a kind of sequential file. Your data elements are written into the record one after another, separated by a carriage return as delimiter—just as they are in sequential files. Also, as in sequential files, numeric data elements are stored as a series of characters rather than as actual numbers.

Thus, when you enter data elements into a record for the first time, DOS sets up the length and position of the fields in the buffer according to the length and order of the elements you write. This information, however, is not stored in the file; consequently, when you go to update the file, unless you have rather exact knowledge of the contents of the record, it becomes difficult to access a particular data element without accessing sequentially all preceding ones.

If you do have such knowledge, DOS gives you the option of using the POSITION command to index into the record a specific number of fields, as well as the B (for byte) parameter to index in a specified number of bytes. Keeping track of this kind of information, however, may be an unreasonable amount of work. Further, since DOS random records are structured like


ODYSSEY:
The Compleat Apventure
NOW IN APPLESOFT*
Synergistic Software
5221 120th Ave. S.E.
Bellevue, Washington 98006
A colorful high-res graphics adventure on land and at sea, in dungeons and castles; different each time you play. Plan your strategy, solve the puzzles, and survive the endless variety of hazards and foes to reap your rewards. Odyssey is an ever fresh mythic adventure.
*Apple II, DOS 3.2 or 3.3, I/A — $30.00.
*Apple II and Applesoft are trademarks of Apple Computer, Inc.

sequential files—with no provision for setting fields larger than the data element itself—the field doesn't allow for the case where a data element enlarges at a later time (dollar amounts in an accounting program, for example). If you want to provide for this situation, you must pad the fields with spaces yourself when you first create the file, or resign yourself to rewriting the rest of the record each time you update a field.

In contrast with Apple DOS, Basic-80 provides full control over field allocation, allowing the programmer to field the buffer any way he chooses, setting a field to enclose single or multiple elements or even a partial element. This process is performed using distinct string variables to represent the fields, so that once buffer fielding has been done, all assigning and updating of elements—as well as all retrieving of elements—is accomplished by a simple reference to the variable used in the field statement. Data elements are placed in the buffer using the SET command either to left-justify (LSET) or right-justify (RSET) the data elements into the field; you control which end of the field will be padded with spaces.

Also, Basic-80 stores numeric values in the file as special tokenized strings that, for a given precision and type of number, are always the same length. The fact that the length of a numeric field remains constant for a given precision and numeric type reduces the amount of times it's necessary to extend or shift field positions as numbers enlarge and shrink. As a result, the programmer is given much more control over the individual file record than is possible under Apple DOS.

To compare actual command forms between DOS and Basic-80, we must first point out that, unlike Applesoft and Integer, Basic-80 has specific disk file commands built in, eliminating the need to precede disk commands with a control-D sequence. Also, since commands exist to set data into the buffer and display data currently in the buffer, it isn't necessary to tell Basic-80 prior to accessing the buffer whether the operation will be a read or a write. In sequential file activity, the mode (either input or output) is set in the OPEN command.

Specific differences, then, are as follows:

First, in sequential file activity, it's not possible in Basic-80 to output to an existing file, since Basic-80 will delete the file automatically before writing any data. To add data to a file, the file must be read in, and simultaneously written out, to a second file with the desired changes and additions. This procedure differs from Apple DOS, which allows both input and output to an existing file. Even DOS, however, forbids data to be inserted into the middle of a sequential file. It's possible to overwrite existing data elements with new ones of exactly the same length, and the file can be extended beyond its current end-of-file mark, but any other write operation will destroy the integrity of the file.

To open a file in Apple DOS thus requires only the statement PRINT "CONTROL-D OPEN FILENAME", while in Basic-80 the mode of the file needs to be stated as well—as, for example, OPEN "I",1,"FILENAME.DAT" for input or OPEN "O",1,"FILENAME.DAT" for output. You'll notice in these Basic-80 commands that a number follows the file mode designator. This is the number of the buffer to be assigned to that file for all activity that follows. Specifying the buffer number allows later commands to omit the file name—for example, CLOSE 1. Writing to or reading from Apple DOS files requires subsequent statements of the form PRINT "CONTROL-D WRITE FILENAME" or . . . READ FILENAME"—followed by statements of the form PRINT "[this is your data]" or INPUT "[this is your data]" respectively, while Basic-80 requires only PRINT #1, "[this is your data]" and INPUT #1, "[this is your data]". Since special print commands are used in Basic-80 to write to the file, normal print statements can be intermixed with file writes. In DOS, on the other hand, any PRINT encountered after the . . . WRITE FILENAME" command will cause a file write unless the file has first been closed.

Finally, Basic-80 allows the writing of formatted numeric data to sequential files by means of the PRINT #X, USING [mask] command, which forces numbers to be written in the

GOTO 33

SOFTCARD Symposium

from page 30

form of the mask used with the statement. PRINT USING is not implemented in Applesoft or Integer.

Specific functions also exist in Basic-80 to detect the end of the sequential file (EOF), the number of disk file sectors read and written since the file was opened (LOC), and the length of the file (LOF). No comparable functions exist in Apple DOS. The remainder of the sequential file commands are, for all intents and purposes, equivalent between the two systems.

With random access files, the same remarks apply about Basic-80's inherent understanding of disk commands and Apple DOS's need for the control-D prefix. In Apple DOS, the file opening command for random files is the same as for sequential files, except for the addition of a record length parameter at the end of the command. The format is PRINT "CONTROL-D OPEN FILENAME,L256". The commands to read and write data to random files are likewise the same as for sequential files, except that the record number in question has to be added to the end of the command—as in PRINT "CONTROL-D WRITE FILENAME,R20". Writing a different record requires you to repeat the entire write command line, with the new record number appended. In Basic-80, the file is opened with the file mode designator as R, telling the system that you wish random access. This command appears in the form OPEN "R",1, "FILENAME.DAT",256. From that point on, reading in a record from disk is accomplished by the statement GET #1,20, where 20 is the record number. Writing a record, similarly, is as easy as PUT #1,20. Gets and puts can be intermixed without special file closure procedures.

In Apple DOS, once the record has been read from disk into the buffer, the writing of data elements into the buffer and reading of data elements from the buffer is accomplished in a manner identical to the writing and reading of elements in sequential files—with print and input statements. In Basic-80, however, there's more to it.

First, the buffer itself must be fielded. This is usually done immediately after the open statement, although it doesn't have to be done then and, in fact, the buffer can be refielded any time. The field statement is the command used to do this, and it takes the form: FIELD #1, X AS A1$, Y AS A2$, and so on, where X and Y are the number of characters you wish in each field. The string variables can be any valid variable name but must not be used on the left side of an = sign anywhere else in the program—except in LSET or RSET statements.

Once the buffer has been fielded, data elements are placed in it, using the LSET and RSET commands as follows: LSET A1$ = "THIS IS YOUR DATA". In this example, the data is placed in the buffer left-justified, and the remainder of the field is padded with space characters. Note that the field variable A1$ must be large enough to contain the data placed in it, or else the data will be truncated. For numeric data, which must first be converted to tokenized strings, the functions MKI$, MKS$, and MKD$ exist. Placing a single-precision variable in the buffer represented by the variable A is done by LSET A1$=MKS$(A). Such a field would normally be only four bytes long, since that's the total space required to store any single-precision variable that's been converted. Integer variables (MKI$) require two bytes, while double-precision variables (MKD$) require eight.

Reading data elements from the buffer is accomplished by treating the field variable name as you would any other variable. Once the record has been read into the buffer with GET, string data can simply be assigned as in FOO$=A1$, or printed as is by PRINT A1$. Numeric data must be converted back from the tokenized string through the use of the functions CVI, CVS, and CVD but are otherwise handled the same way as strings, as in FOO=CVS(A1$) or PRINT CVS(A1$). Field variable names can be referenced, accessed, and printed in any order and at any time later in the program—unless they are changed by subsequent gets or other field statements. The increased flexibility gained through these techniques more than makes up for any additional time you have to spend becoming proficient in them.

While we have done no more than scratch the surface of sequential and random file handling, we have had the opportunity to look at the logical and physical structures involved, as well as the commands. We hope this information will help make your use of text files more efficient and productive. Until next month . . .

THE PASCAL PATH

By Jim Merritt

Tools of the Craft, Part 5

Last time, I promised that we would not only conclude our discussion of programmed decision making by examining Pascal's CASE statement, but also dig deeper into WriteLn and other miscellaneous areas of the language. Unfortunately, space and time do not permit me to accomplish both goals this month, so I'm postponing the discussion of CASE (and its relationship to the IF-THEN-ELSE statement) until I can give it the space and attention it deserves. Instead, this installment of the Pascal Path will deal with several less weighty matters, all of which nevertheless figure prominently in the writing of nearly every useful program.

More about Output. *WriteLn Can Display a List of Items.* From the start, we have been using WriteLn to display data on the console output device. For example, the statement

```
WriteLn(8+8)
```

causes the value 16 to be displayed on the console. After the value is displayed, the console's cursor is positioned at the beginning of the next output line. In the examples I've shown you so far, I've displayed exactly one item per WriteLn. However, it's possible to list several items between a WriteLn's parentheses, using commas to separate the items. Each of the values will be displayed, one after another, on the same output line. Here's a WriteLn statement that displays three values at once:

```
WriteLn(1,2,3);
```

When the statement above is executed, the following output appears on the console:

```
123
```

This probably isn't what you'd expect (or want) to see. On the other hand, WriteLn is doggedly consistent in its bunching-together—its *concatenation*—of display items on the same line. You can count on this regular behavior, and work around it quite easily. To emphasize that three different numbers are being displayed, you must tell the computer to put blanks (or some other separating punctuation) between them, like so:

```
WriteLn(1,' ',2,' ',3);
```

Here's the display that's generated when the revised WriteLn statement is executed:

```
1 2 3
```

It's up to you to separate distinct data items from one another on a display line by including explicit blanks or other characters at appropriate points in the parenthetical data list. In general, Pascal gives you total freedom to display data any way you choose, but it requires that you be responsible for every detail of your display. (Didn't someone once say the cost of freedom is eternal vigilance?)

Output Formatting: WriteLn Field-Width Specifications. Programs, especially those used in business and engineering, often display data in columnar form. Try compiling and executing this example:

```
PROGRAM
  Columns;
BEGIN
  WriteLn(1,' ',10,' ',100);
  WriteLn(1000,' ',10000,' ',0);
  WriteLn;
  WriteLn(1:5,' ',10:5,' ',100:5);
  WriteLn(1000:5,' ',10000:5,' ',0:5);
END.
```

The first two lines of output data are displayed in Pascal's usual *free-format.* Since the blank spaces between data items are fixed in size, and the data items themselves are of different sizes, the items in the first line fail to line up with those in the second. Computer professionals might say, "The output is poorly *formatted.*" Items in the fourth and fifth output lines, however, are synchronized into columns; they are "more pleasantly formatted." This is because a *field-width specification* was appended to each numeric data item.

A field-width specification consists of a colon, followed by an integer expression. The expression evaluates to the *minimum* number of character positions that a particular data item should occupy on the output line. This means that, if the item, as displayed, is *shorter* than the specified field-width, Pascal will make up the difference with blanks (positioned either before or after the item, depending upon the item's type). However, when the data item is *longer* than the given field-width, Pascal still displays all of it, ignoring the field-width specification. For instance, the number 1000 cannot be displayed in less than four character positions. The following program demonstrates the misalignment that can occur when a field-width specification is too small for the data item to be displayed;

```
PROGRAM
  MisAlign;
 CONST
 FW= 4;
BEGIN
  WriteLn(1:FW, 10:FW, 100:FW);
  WriteLn(75:FW, 49:FW, 20:FW);
  WriteLn(10:FW, 1000:FW, 1:FW);
  WriteLn(10000:FW, 50:FW, 60:FW);
  WriteLn(0:FW, 5:FW, 10:FW);
  WriteLn(200:FW, 250:FW, 999:FW);
END.
```

Here's MisAlign's output:

```
   1  10 100
  75  49  20
  101000   1
 10000  50  60
   0   5  10
 200 250 999
```

The first, second, fifth, and sixth output lines are in perfect columnar form, with the numbers aligned on their right-hand sides. Notice there was no need to include explicit blanks in the WriteLn statements in order to separate the numbers on each line. Since no number is longer than three digits, and the minimum field width for each is 4 (the value of the constant FW), at least one blank is guaranteed to appear at the beginning of each field.

The third and fourth output lines, however, show signs of misalignment. In the third line, the first two numbers run together, because the second number, 1000, requires four character positions to display, and the field-width is exactly four. Since the data item is not shorter than the minimum field width, Pascal doesn't have to fill the unused character positions with blanks. Thus, the second number on the third output line begins where the first number ends, and you see what appears to be one huge number. In the fourth line, the number 10000 is one digit larger than the minimum field width. Nevertheless, Pascal displays the entire number, in a field whose actual width is now one larger than that intended by the programmer. Subsequent fields are thus thrown out of alignment.

Take Care When Determining Field Widths. When specifying field widths, you should make a special effort to tailor

them for the types of data to be displayed, and this means that you have to think carefully about the properties of the data involved. For instance, when you are figuring field widths for numbers, remember to include room for the minus sign, if the item can be negative, as well as the decimal point, letter *e*, and exponent (including its sign) when you're dealing with a Real number. (If it's any help to you, I generally use 12 as the field width for Real items, and 6 for Integers. Do you see why?)

Field-width specifications apply (and may be appended) not only to numbers, but also to Char and String data items. If such specification is appended, Apple Pascal assumes the field width to be the exact size of the data item. Pascal behaves differently when the displayed form of a data item is shorter than the field width specified for it, depending upon whether the item is numeric, Char, or String. As you've seen, numeric items are *right aligned*, because Pascal makes up any unused character positions by *preceding* the data item with the appropriate number of blanks. Char and String values, however, are *left aligned*. If a String or Char item is shorter than the specified field width, Pascal first displays it, then *follows* with the number of blanks necessary to fill any unused character positions.

Real Data Can Have Two Field Specifications. A subsidiary field specification may be appended to Real items, behind the usual minimum field-width expression. It also consists of a colon followed by an Integer expression but tells Pascal the number of decimal places that should be displayed for the particular Real number. (If no subsidiary field specification is given, Pascal will show all significant figures that lie beyond the decimal point. Note that you cannot specify decimal precision without also specifying minimum field width.)

The following statement displays the contents of RealVariable, with a precision of three digits beyond the decimal point, in a field that is at least twelve positions wide:

```
WriteLn(RealVariable:12:3);
```

If RealVariable contains the approximate value of Pi (3.14159 . . .) when this WriteLn is executed, here is the output that will be produced:

~~bbbbbbb~~3.142

The *b* symbol stands for a blank, which is, of course, invisible. I use the special symbol here so that you can count the number of *leading blanks* that Pascal generates to fill the given field, and so satisfy yourself that the field-width specification is, in fact, enforced.

Did you notice that Pascal rounds the displayed output to whatever precision is requested? Naturally, even though 3.142 is displayed, the value contained in RealVariable, 3.14159, remains unaltered.

WriteLn's Fraternal Twin: Write. The console's cursor or print head is always repositioned to the start of the next output line whenever WriteLn is executed. This is not always desirable. Let's suppose, for example, that you want to fill an output line with asterisks. The only way you now know to do this is to use a single WriteLn statement that displays a string literal which contains as many asterisks as are necessary to fill a line on your terminal. For the standard Apple screen, forty columns wide, you would use

```
WriteLn(' ******************************');
```

The statement above, of course, will not fill an eighty-character screen; to do that, you must actually increase the length of the literal. Such program modifications are prone to error; you must count every asterisk in the literal to make sure the correct number of characters will be displayed, and you can easily make a mistake—especially since typing and counting all the asterisks is very boring work.

Clever as you are, you have probably decided already that it would be better to use a loop to display one asterisk at a time—one after another on the same line—until the proper number of them are on the screen. Tearing into the problem, you might write the following program in Pascal:

```
PROGRAM
  Asterisks;

  CONST
   LineLength= 40;

  VAR
   NumChars
    :Integer;
BEGIN
  FOR NumChars := 1 TO LineLength DO
   WriteLn('*');
END.
```

Unfortunately, instead of showing asterisks in the horizontal direction, your computer would display them vertically, down the left side of the screen, since WriteLn would force the start of a new output line after every repetition.

To get around the difficulty and accomplish the task, you can use *Write*. WriteLn and Write are nearly identical. Each can include a list of data items between parentheses; any of the items in a Write or WriteLn data list may be suffixed with a field-width specification. The only differences between the two are that WriteLn forces any subsequent output to start at the beginning of the next output line, while Write does not; and WriteLn may be used with or without a parenthetical data list, while Write must always include a data list.

Note that WriteLn can be simulated with Write statements. For instance, the statement

```
WriteLn(A,B,C)
```

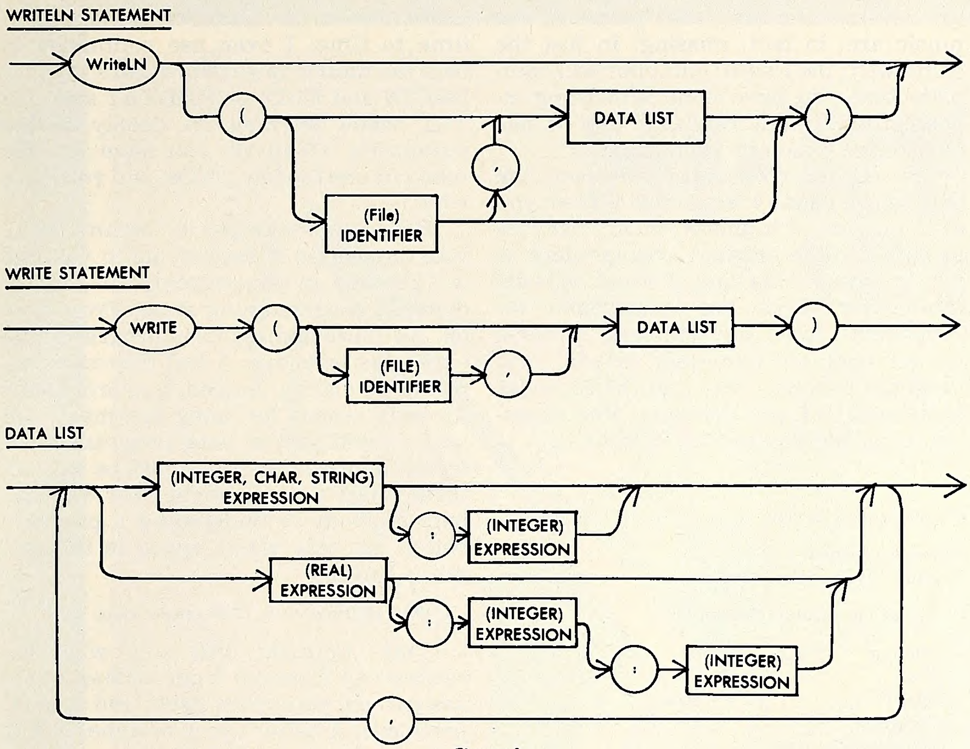


Figure 1.

is equivalent to the following sequence of statements:

```
Write(A);
Write(B);
Write(C);
WriteLn;
```

For those who absolutely have to see Write and WriteLn expressed as railroad diagrams, look no further than figure 1. Notice that, for either, the data list may be preceded by an optional *file identifier,* which we haven't used yet. I won't go into the significance of the file identifier now, except to say that its absence, in general, causes data to be displayed on the console device. Soon you will use this optional element of the Write and WriteLn statements to redirect output to other devices, such as a printer or disk file. How to do this, though, is clearly the subject of a future column.

Comments. Pascal programs may include *comments* at almost any point. A comment is an arbitrarily long memorandum, written by the programmer to suit his or her own purposes, and is treated by the compiler as a single blank (but otherwise ignored). Thus, you may place a comment anywhere a blank may be used (except in the midst of quoted Char or String literals). Every comment begins with a certain delimiting symbol and ends with a corresponding symbol. Usually these are the braces, { and }. Apple III computers include keyboards that can generate the braces—and video hardware that can display them—but unmodified Apple IIs do not. (In fact, the braces *are* displayed on a standard Apple II screen, but confusingly, as the square-brackets, [and], which are used for entirely different purposes in Pascal! This does *not* mean, however, that you can type square brackets from the Apple II keyboard and expect them to be treated as braces. On the display, brackets and braces are both shown as brackets, but from the keyboard, it's impossible to type a brace, and any brackets you type will be treated as brackets. Any attempt you make to enclose a comment in square brackets will cause the compiler to complain.)

How To Get By If You Can't Brace Yourself. To accommodate computer systems that, for one reason or another, do not have braces, Pascal permits comments to begin with the couplet (* and end with the couplet *). Since I assume you have only a standard Apple II, I will use only the second kind of comment delimiters in my examples. To familiarize you with comments, though, let me present a few instances of each kind:

Correct Comments

```
{This is a comment.}
(* So is this *)
{ One kind of comment (* can include the other *)! }
(* It works { both ways } *)
(* Comments are permitted
   to span
   several lines. These
   five lines make up only
   one single comment. *)
```

Incorrect Comments

```
{ You can't close a "brace-comment"
  with a "star-paren", *)
(* Or vice-versa }
(* The closing delimiter must not occur *) in the
   midst of the comment *)
(* This means that you can't "nest" (* comments
   of the same kind *) *)
```

Beware! Unclosed Comments Eat Programs! Because comments can span multiple lines in the program, it's easy to forget to close a long comment with the proper delimiter. If you fail to close a comment, the Pascal compiler will keep bypassing your program text until either it reaches the end of the text file, or it finally *does* encounter the proper closing delimiter, in some later comment. In the first case, you will receive notice that error #401 ("Unexpected end of input") has occurred. What this error message really means is that the compiler has reached the end of the source file without recognizing the final END keyword that is the official termination signal for all compilations. Even if the keyword is there, in the proper place, the compiler can't recognize it, because it skips past text as fast as it can, trying to find the end of the unclosed comment.

The second case, where the closing delimiter in one comment is recognized as terminating a previous, unclosed comment, may lead the compiler to produce some incredibly bizarre error messages that don't even begin to hint at the true cause of the problem. This phenomenon is similar to the confusion you feel when listening to a musical recording that has been edited badly, without your knowledge. The music just doesn't fit together correctly, and you might say to yourself, "the beat is off," or "the key shifts weirdly," without realizing that pieces of the music are, in fact, missing. In just the same way, the Pascal compiler will complain that you have done something incorrectly, without realizing that it has overlooked much of your program!

In musical recordings, however, the editing is usually so smooth that you aren't aware of it unless you are very familiar with the original arrangement or performance. This kind of situation is desirable for music but catastrophic for programs! It is, nevertheless, possible for an unclosed comment to edit your program in such a way that the compiler is unaware of any problem. For example, consider this section of code:

```
(* Initialization Section

Retaliate := False;
LimitedStrike := False;

(* End of Initialization Section *)

IF Retaliate
 THEN
  BEGIN
   OpenSilo;
   PlotMissileCourse;
   LaunchMissile;
   IF LimitedStrike
    THEN
     ArmWarHead(Neutron) (* Kill Personnel Only *)
    ELSE
     ArmWarHead(Hydrogen); (* Devastate Target *)
   SendEBSMessage; (* We'll be the last to know *)
  END;
```

Because the introductory comment isn't closed, the entire initialization section is ignored, effectively edited out of the program! Thus, the values of Retaliate and LimitedStrike are, for all intents and purposes, random garbage. Unfortunately, the "random garbage" might be interpreted as "True" or "False," and a global catastrophe could result. Usually, the only way problems like these are discovered is to compile and use the program in question, then track down the bug when funny behavior becomes apparent (which may not happen for days, weeks, months, or years!).

As long as you are careful to close your comments, you will have none of the problems I've mentioned here (and may even live to a ripe old age). On the other hand, if you do start getting strange error messages or observe weird program behavior, perhaps an unclosed comment is the culprit.

How and Why Comments Are Used. Sometimes comments show who wrote or modified a program, as well as what was accomplished and when it was finished. (This is usually called the program's *maintenance history.*) Most programmers use comments to point out a program's primary functions, limitations, and key parts, or to highlight and introduce important sections of a program. Comments should always be used to clarify the obscure methods and advanced programming techniques that you will no doubt discover and use from time to time. I even use comments to help me match up keyword pairs, such as BEGIN and END, or REPEAT and UNTIL, when working on deeply nested structures. (Wait till you have one too many or one too few ENDs, and you'll see why!)

How you make use of comments in your programs is entirely up to you and will become an important part of your individual programming style. From now on, I will use comments in programming examples whenever I feel they can contribute to clarity. Indeed, I believe that's the only reason for using comments—to make some part of your program more understandable than it would be without them. Here's a comment that wouldn't help anybody to understand a program better but only waste space in the program listing:

```
Hour := Hour + 1; (* Increment Hour *)
```

This comment tells only what the reader can discover from looking at the assignment statement itself, and so is superfluous. A better use of comments is to tell *why* something is done:

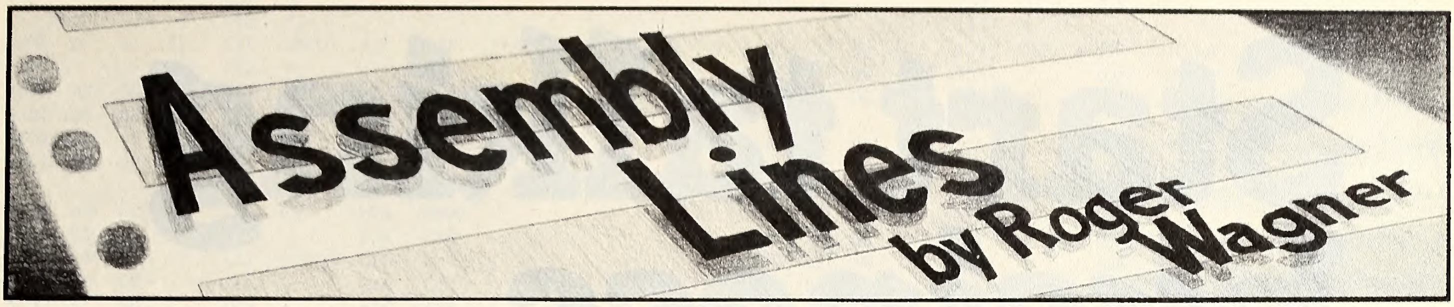
```
Hour := Hour + 1; (* Spring forward to
                     daylight-saving time *)
```

One other thing: Just because you write a program for your own personal use and don't expect anyone else to use or read it, don't be fooled into thinking that commenting your program is unnecessary. I'm certainly no saint, even though I presume to tell *you* how to program. There have been many times when I have programmed something in a hurry and have not felt justified in commenting my work. I can't think of a single case in which I was ultimately happy with such a decision. You *always* have some use for old programs; either you can rewrite them to suit new purposes and situations that you couldn't have foreseen when doing the original work, or you can extract techniques—and sometimes even entire pages of statements, taken verbatim—for the benefit of new programs.

If you haven't worked with a program for a while, you may have only the vaguest memory of its design details, or the subtleties of behavior that it might exhibit from time to time, yet you need this information in order to be successful at adapting or rewriting the program. It's tedious, and usually unnecessary, to refresh your memory by running a program, or by following the listing and simulating its operation in your mind. Still, this is what you have to do if a program doesn't include enough comments to let you know what is really going on. Consider how much worse things are for a total stranger to pick up your uncommented program and try to make sense of it!)

Knowing when to comment, and what to say in a comment, is difficult. After all,

GOTO 184



Everyone's Guide to Assembly Language, Part 14

Commands Covered So Far:

JMP	LDA	LDX	LDY	TAX
JSR	STA	STX	STY	TAY
RTS	INC	INX	INY	TXA
NOP	DEC	DEX	DEY	TYA
—	CMP	CPX	CPY	PHA
BEQ	BNE	BCC	BCS	PLA
SEC	CLC	ADC	SBC	

This month will challenge your devotion to the cause of learning assembly language programming. Until now the source listings have been very short and easily typed in a few minutes' time. Unfortunately, the listings for this installment are a bit longer and will require an assembler. But chin up! The result will be worth it! I've received quite a number of requests for information on how to read and write files on the disk. The programs listed will combine many of the techniques and routines you've learned so far into a single mini-data-base program.

The first program saves and loads the data by means of a simple Bsave/Bload operation. This is fast and straightforward. Here's the listing:

```
                 ***************************
                 * NAME FILE DEMO PROGRAM *
                 ***************************
                 *
                 *
                          OBJ   $6000
                          ORG   $6000
                 *
                 HOME     EQU   $FC58
                 COUT     EQU   $FDED
                 RDKEY    EQU   $FD0C
                 GETLN    EQU   $FD75
                 BUFF     EQU   $200
                 VTAB     EQU   $FC22
                 CH       EQU   $24
                 CV       EQU   $25
                 CTR      EQU   $08
                 PTR      EQU   $06
                 *
                 *
6000: A9 00      ENTRY    LDA   #$00
6002: 85 06               STA   PTR
6004: A9 10               LDA   #$10
6006: 85 07               STA   PTR+1
6008: A9 B1               LDA   #$B1
600A: B5 08               STA   CTR
                 *
600C: A0 00      CLR      LDY   #$00
600E: 91 06               STA   (PTR),Y
6010: C8                  INY
6011: A9 A0               LDA   #$A0
6013: 91 06               STA   (PTR),Y
6015: A9 00               LDA   #$00
6017: C8                  INY
6018: 91 06               STA   (PTR),Y
601A: E6 07               INC   PTR+1
601C: E6 08               INC   CTR
601E: A5 08               LDA   CTR
6020: C9 B6               CMP   #$B6
6022: 90 E8               BCC   CLR
                 *
                 *        PUTS '#1-5,SPC,00' IN BUFFER
                 *
6024: 20 5B FC   MENU     JSR   HOME
6027: A9 02      P1       LDA   #$02
6029: 85 25               STA   CV ; VTAB 3
602B: 20 22 FC            JSR   VTAB
602E: 20 C0 61            JSR   PRINT
6031: B1 A9 A0            ASC   "1) INPUT NAMES"
603F: BD 00               HEX   8D00
                 *
6041: A9 04      P2       LDA   #$04
6043: 85 25               STA   CV
6045: 20 22 FC            JSR   VTAB ; VTAB 5
604B: 20 C0 61            JSR   PRINT
604B: B2 A9 A0            ASC   "2) PRINT NAMES"
6059: 8D 00               HEX   8D00
                 *
605B: A9 06      P3       LDA   #$06
605D: B5 25               STA   CV
605F: 20 22 FC            JSR   VTAB ; VTAB 5
6062: 20 C0 61            JSR   PRINT
6065: B3 A9 A0            ASC   "3) SAVE NAMES"
6072: 8D 00               HEX   8D00
                 *
6074: A9 08      P4       LDA   #$08
6076: B5 25               STA   CV
6078: 20 22 FC            JSR   VTAB ; VTAB 9
607B: 20 C0 61            JSR   PRINT
607E: B4 A9 A0            ASC   "4) LOAD NAMES"
60BB: BD 00               HEX   8D00
                 *
60BD: A9 0A      P5       LDA   #$0A
60BF: 85 25               STA   CV
6091: 20 22 FC            JSR   VTAB ; VTAB 11
6094: 20 C0 61            JSR   PRINT
6097: B5 A9 A0            ASC   "5) END PROGRAM"
60A5: BD 00               HEX   8D00
                 *
60A7  A9 0C      P6       LDA   #$0C
60A9: B5 25               STA   CV
60AB: 20 22 FC            JSR   VTAB ; VTAB 13
60AE: 20 C0 61            JSR   PRINT
60B1: D7 C8 C9            ASC   "WHICH DO YOU WANT? "
60C4: 00                  HEX   00
                 *
60C5: 20 0C FD   M1       JSR   RDKEY
60CB: C9 B1               CMP   #$B1 ; '1'
60CA: D0 06               BNE   M2
60CC: 20 FB 60            JSR   INPUT
60CF: 4C 24 60            JMP   MENU
60D2: C9 B2      M2       CMP   #$B2 ; '2'
60D4: D0 09               BNE   M3
60D6: 20 40 61            JSR   DSPLY
60D9: 20 0C FD            JSR   RDKEY
60DC: 4C 24 60            JMP   MENU
60DF: C9 B3      M3       CMP   #$B3 ; '3'
60E1: D0 06               BNE   M4
60E3: 20 76 61            JSR   SAVE
60E6: 4C 24 60            JMP   MENU
60E9: C9 B4      M4       CMP   #$B4 ; '4'
60EB: D0 06               BNE   M5
60ED: 20 9E 61            JSR   LOAD
60F0: 4C 24 60            JMP   MENU
60F3: C9 B5      M5       CMP   #$B5 ; '5'
60F5: D0 01               BNE   M6
60F7: 60                  RTS
60FB: 4C 24 60   M6       JMP   MENU
```

Start talking business

with your Apple

COBOL is the most effective business language.
Apple II is the most friendly business computer.
CIS COBOL with FORMS-2 brings together the best features of COBOL and Apple to enable you to deliver the most effective, user-friendly applications.

Business Programmers: Take the COBOL expertise you have acquired on big business mainframes, and use it on Apple II to create friendly applications that will talk directly to your users – where it suits them best, on their own desks.

CIS COBOL's dynamic module loading gives you big application capability and the FORMS-2 source generator lets you build and modify conversational programs from visual screen formats, creating much of the code automatically.

Application vendors: CIS COBOL with FORMS-2 steps up the pace for your development of the high quality professional application packages needed today. And creating them in COBOL makes them more maintainable.

Over half the Apple II's now being sold are going to business or professional users so demand for quality applications is growing fast, creating big business opportunities for you.

Stability proven by the US Government. CIS COBOL has been tested and approved for two consecutive years by the US General Services Administration as conforming to the ANSI '74 COBOL Standard. Apple II under CP/M is included in CIS COBOL's 1981 GSA Certificate of Validation (at Low-Intermediate Federal Standard plus Indexed I-O and Level 2 Inter-Program Communication).

Get your hands on CIS COBOL at your Apple dealer.
Talk business with him now!

Micro Focus Inc., 1601 Civic Center Drive
Santa Clara, CA 95050. Phone: (408) 248-3982.

MICRO FOCUS

CIS COBOL with FORMS-2 for use on the Apple II with CP/M is an Apple Distributed Product.
CIS COBOL and FORMS-2 are trademarks of Micro Focus. CP/M is a trademark of Digital Research. Apple II is a trademark of Apple Computer.

```
                 *
                 *
60FB: 20 40 61   INPUT  JSR  DSPLY ; SHOW WHAT'S THERE
                 *
60FE: A9 00      I0     LDA  #$00
6100: 85 06             STA  PTR
6102: A9 10             LDA  #$10
6104: 85 07             STA  PTR+1 ; SET PTR = $1000
                 *
6106: A9 00      @@     LDA  #$00
6108: 85 08             STA  CTR
610A: 18         ILOOP  CLC
610B: A5 08             LDA  CTR
610D: 65 08             ADC  CTR
610F: 85 25             STA  CV
6111: 20 22 FC          JSR  VTAB
6114: A9 00             LDA  #$00
6116: 85 24             STA  CH
6118: A8                TAY
6119: 20 27 61          JSR  IP
611C: E6 07             INC  PTR+1
611E: E6 08             INC  CTR
6120: A9 04             LDA  #$04
6122: C5 08             CMP  CTR
6124: B0 E4             BCS  ILOOP ; GET 5 NAMES
                 *
6126: 60         IFIN   RTS
                 *
6127: A2 00      IP     LDX  #$00
6129: 20 75 FD          JSR  GETLN
612C: 8A                TXA
612D: F0 10             BEQ  IPFIN ; EXIT IF <CR> ONLY
612F: A8                TAY
6130: A9 00             LDA  #$00
6132: 99 00 02          STA  BUFF,Y;PUT '0' AT END
6135: B9 00 02   IPLOOP LDA  BUFF,Y
6138: 91 06             STA  (PTR),Y ; MOVE DATA TO PTR
                 *           ; BLOCK.
613A: 88                DEY
613B: C0 FF             CPY  #$FF
613D: D0 F6             BNE  IPLOOP
613F: 60         IPFIN  RTS
                 *
6140: 20 58 FC   DSPLY  JSR  HOME
6143: A9 00             LDA  #$00
6145: 85 08             STA  CTR
                 *
6147: 85 06             STA  PTR
6149: A9 10             LDA  #$10
614B: 85 07             STA  PTR+1
614D: 18         D0     CLC
614E: A5 08             LDA  CTR
6150: 65 08             ADC  CTR
6152: 85 25             STA  CV ; VTAB (2*CTR)+1
6154: 20 22 FC          JSR  VTAB
6157: A9 00             LDA  #$00
6159: 85 24             STA  CH ; HTAB 1
615B: A8                TAY
                 *
615C: B1 06      D1     LDA  (PTR),Y
615E: F0 06             BEQ  D1FIN
6160: 20 ED FD          JSR  COUT
6163: C8                INY
6164: D0 F6             BNE  D1 (ALWAYS)
                 *
6166: A9 8D      D1FIN  LDA  #$8D
6168: 20 ED FD          JSR  COUT ; END W/ <CR>
616B: E6 07             INC  PTR+1
616D: E6 08             INC  CTR
616F: A9 04             LDA  #$04
6171: C5 08             CMP  CTR
6173: B0 D8             BCS  D0 ;PRINT 5 NAMES
                 *
6175: 60         DSFIN  RTS
                 *
                 *
6176: A9 8D      SAVE   LDA  #$8D
6178: 20 ED FD          JSR  COUT ; CLEAR OUTPUT BUFFER
617B: 20 C0 61   OPEN   JSR  PRINT
617E: 84                HEX  84
617F: C2 D3 C1          ASC  "BSAVE DEMO FILE,A$1000,L$500"
619B: 8D 00             HEX  8D00
```

```
                     *
619D:  60            SFIN      RTS
                     *
                     *
619E:  A9  8D        LOAD      LDA    #$8D
61A0:  20  ED  FD              JSR    COUT
                     *
61A3:  20  C0  61              JSR    PRINT
61A6:  84                      HEX    84
61A7:  C2  CC  CF              ASC    "8LOAD DEMO FILE,A$1000"
61BD:  8D  00                  HEX    8D00
                     *
61BF:  60                      RTS
                     *
                     *
                     *
61C0:  68            PRINT     PLA
61C1:  85  06                  STA    PTR
61C3:  68                      PLA
61C4:  85  07                  STA    PTR+1
61C6:  A0  01                  LDY    #$01
61C8:  B1  06        P0        LDA    (PTR),Y
61CA:  F0  06                  BEQ    PFIN
61CC:  20  ED  FD              JSR    COUT
61CF:  C8                      INY
61D0:  D0  F6                  BNE    P0 (ALWAYS)
                     *
61D2:  18            PFIN      CLC
61D3:  98                      TYA
61D4:  65  06                  ADC    PTR
61D6:  85  06                  STA    PTR
61D8:  A5  07                  LDA    PTR+1
61DA:  69  00                  ADC    #$00
61DC:  48                      PHA
61DD:  A5  06                  LDA    PTR
61DF:  48                      PHA
61E0:  60            EXIT      RTS
                     *
                     *
61E1:  00            EOF       BRK
                     *
                     *
```

```
                     ******************************
                     *NAME FILE DEMO PROGRAM #2 *
                     ******************************
                     *
                     *
                               O8J    $6000
                               ORG    $6000
                     *
                     HOME      EQU    $FC58
                     COUT      EQU    $FDED
                     RDKEY     EQU    $FD0C
                     GETLN     EQU    $FD75
                     8UFF      EQU    $200
                     VTA8      EQU    $FC22
                     CH        EQU    $24
                     CV        EQU    $25
                     CTR       EQU    $08
                     PTR       EQU    $06
                     *
                     PROMPT    EQU    $33
                     CURLIN    EQU    $75
                     LANG      EQU    $AA86
                     REENTRY   EQU    $3D0
                     *
6000:  A9  40        ENTRY     LDA    #$40
6002:  8D  B6  AA              STA    LANG ; LANG = FP
6005:  85  76                  STA    CURLIN+1 ; RUNNING PROG
6007:  A9  06                  LDA    #$06
6009:  85  33                  STA    PROMPT ; NOT DIRECT MODE
6008:  A9  00                  LDA    #$00
600D:  85  06                  STA    PTR
600F:  A9  10                  LDA    #$10
6011:  85  07                  STA    PTR+1
6013:  A9  B1                  LDA    #$B1
6015:  85  08                  STA    CTR
                     *
6017:  A0  00        CLR       LDY    #$00
6019:  91  06                  STA    (PTR),Y
601B:  C8                      INY
601C:  A9  A0                  LDA    #$A0
601E:  91  06                  STA    (PTR),Y
6020:  A9  00                  LDA    #$00
6022:  C8                      INY
6023:  91  06                  STA    (PTR),Y
6025:  E6  07                  INC    PTR+1
6027:  E6  08                  INC    CTR
6029:  A5  08                  LDA    CTR
602B:  C9  B6                  CMP    #$86
602D:  90  E8                  8CC    CLR
                     *
                     *                PUTS '#1-5,SPC,00' IN 8UFFER
                     *
602F:  20  58  FC    MENU      JSR    HOME
6032:  A9  02        P1        LDA    #$02
6034:  85  25                  STA    CV ; VTAB 3
6036:  20  22  FC              JSR    VTA8
6039:  20  0A  62              JSR    PRINT
603C:  B1  A9  A0              ASC    "1) INPUT NAMES"
604A:  8D  00                  HEX    8D00
                     *
604C:  A9  04        P2        LDA    #$04
604E:  85  25                  STA    CV
6050:  20  22  FC              JSR    VTA8 ; VTAB 5
6053:  20  0A  62              JSR    PRINT
6056:  B2  A9  A0              ASC    "2) PRINT NAMES'
6064:  8D  00                  HEX    8D00
                     *
6066:  A9  06        P3        LDA    #$06
6068:  85  25                  STA    CV
606A:  20  22  FC              JSR    VTAB ; VTA8 7
606D:  20  0A  62              JSR    PRINT
6070:  B3  A9  A0              ASC    "3) SAVE NAMES"
607D:  8D  00                  HEX    8D00
                     *
607F:  A9  08        P4        LDA    #$08
6081:  85  25                  STA    CV
6083:  20  22  FC              JSR    VTA8 ; VTA8 9
6086:  20  0A  62              JSR    PRINT
6089:  B4  A9  A0              ASC    "4) LOAD NAMES"
6096:  8D  00                  HEX    8D00
                     *
6098:  A9  0A        P5        LDA    #$0A
609A:  85  25                  STA    CV
609C:  20  22  FC              JSR    VTAB ; VTAB 11
609F:  20  0A  62              JSR    PRINT
60A2:  B5  A9  A0              ASC    "5) END PROGRAM"
60B0:  8D  00                  HEX    8D00
                     *
6082:  A9  0C        P6        LDA    #$0C
60B4:  85  25                  STA    CV
60B6:  20  22  FC              JSR    VTAB ; VTAB 13
60B9:  20  0A  62              JSR    PRINT
60BC:  D7  C8  C9              ASC    "WHICH DO YOU WANT? "
60CF:  00                      HEX    00
                     *
60D0:  20  0C  FD    M1        JSR    RDKEY
60D3:  C9  B1                  CMP    #$81 ; '1'
60D5:  D0  06                  BNE    M2
60D7;  20  08  61              JSR    INPUT
60DA:  4C  2F  60              JMP    MENU
60DD:  C9  B2        M2        CMP    #$B2 ; '2'
60DF:  D0  09                  BNE    M3
60E1:  20  4D  61              JSR    DSPLY
60E4:  20  0C  FD              JSR    RDKEY
60E7:  4C  2F  60              JMP    MENU
60EA:  C9  83        M3        CMP    #$83 ; '3'
60EC:  D0  06                  8NE    M4
60EE:  20  83  61              JSR    SAVE
60F1:  4C  2F  60              JMP    MENU
60F4:  C9  84        M4        CMP    #$84 ; '4'
60F6:  D0  06                  8NE    M5
60F8:  20  C7  61              JSR    LOAD
60FB:  4C  2F  60              JMP    MENU
60FE:  C9  85        M5        CMP    #$B5 ; '5'
6100:  D0  03                  8NE    M6
6102:  4C  D0  03              JMP    REENTRY
6105:  4C  2F  60              JMP    MENU
                     *
                     *
```

```
6108: 20  4D  61   INPUT    JSR  DSPLY ; SHOW WHAT'S THERE
                   *
610B: A9  00       IO       LDA  #$00
610D: 85  06                STA  PTR
610F: A9  10                LDA  #$10
6111: 85  07                STA  PTR+1 ; SET PTR = $1000
                   *
6113: A9  00       @@       LDA  #$00
6115: 85  08                STA  CTR
6117: 18           ILOOP    CLC
6118: A5  08                LDA  CTR
611A: 65  08                ADC  CTR
611C: 85  25                STA  CV
611E: 20  22  FC            JSR  VTAB
6121: A9  00                LDA  #$00
6123: 85  24                STA  CH
6125: A8                    TAY
6126: 20  34  61            JSR  IP
6129: E6  07                INC  PTR+1
612B: E6  08                INC  CTR
612D: A9  04                LDA  #$04
612F: C5  08                CMP  CTR
6131: B0  E4                BCS  ILOOP; GET 5 NAMES
                   *
6133: 60           IFIN     RTS
                   *
6134: A2  00       IP       LDX  #$00
6136: 20  75  FD            JSR  GETLN
6139: 8A                    TXA
613A: F0  10                BEQ  IPFIN ; EXIT IF <CR> ONLY
613C: A8                    TAY
613D: A9  00                LDA  #$00
613F: 99  00  02            STA  BUFF,Y : PUT '0' AT END
6142: B9  00  02   IPLOOP   LDA  BUFF,Y
6145: 91  06                STA  (PTR),Y ;MOVE DATA TO PTR
                   *; BLOCK.
6147: 88                    DEY
6148: C0  FF                CPY #$FF
614A: D0  F6                BNE IPLOOP
614C: 60           IPFIN    RTS
                   *
614D: 20  58  FC   DSPLY    JSR HOME
6150: A9  00                LDA  #$00
6152: 85  08                STA  CTR
                   *
6154: 85  06                STA  PTR
6156: A9  10                LDA  #$10
6158: 85  07                STA  PTR+1
615A: 18           DO       CLC
615B: A5  08                LDA  CTR
615D: 65  08                ADC  CTR
615F: 85  25                STA  CV ; VTAB (2*CTR)+1
6161: 20  22  FC            JSR  VTAB
6164: A9  00                LDA  #$00
6166: 85  24                STA  CH ; HTAB 1
6168: A8                    TAY
                   *
6169: B1  06       D1       LDA  (PTR),Y
616B: F0  06                BEQ  D1FIN
616D: 20  ED  FD            JSR  COUT
6170: C8                    INY
6171: D0  F6                BNE  D1 (ALWAYS)
                   *
6173: A9  8D       D1FIN    LDA  #$8D
6175: 20  ED  FD            JSR  COUT ; END w/ <CR>
6178: E6  07                INC  PTR+1
617A: E6  08                INC  CTR
617C: A9  04                LDA  #$04
617E: C5  08                CMP  CTR
6180: B0  D8                BCS  DO ; PRINT 5 NAMES
                   *
6182: 60           DSFIN    RTS
                   *
                   *
6183: A9  8D       SAVE     LDA  #$8D
6185: 20  ED  FD            JSR  COUT ; CLEAR OUTPUT
6188: 20  0A  62   OPENW    JSR  PRINT
618B: 84                    HEX  84 ; CTRL-D
618C: CF  D0  C5            ASC  "OPEN NAME TEXT FILE"
619F: 8D  84                HEX  8D84
61A1: D7  D2  C9   WRITE    ASC  "WRITE NAME TEXT FILE"
61B5: 8D  00                HEX  8D00
                   *
61B7: 20  4D  61   SVLOOP   JSR  DSPLY ; PRINT NAMES TO DISK
                   *
61BA: 20  0A  62   CLOSEW   JSR  PRINT
61BD: 8D  84                HEX  8D84
```

```
61BF: C3  CC  CF   207                 ASC   "CLOSE"
61C4: 8D  00       208                 HEX   8D00
61C6: 60           209   SVFIN         RTS
                   210   *
                   211   *
61C7: A9  8D       212   LOAD          LDA   #$8D
61C9: 20  ED  FD   213                 JSR   COUT
                   214   *
61CC: 20  0A  62   215   OPENR         JSR   PRINT
61CF: 84           216                 HEX   84
61D0: CF  D0  C5   217                 ASC   "OPEN NAME TEXT FILE"
61E3: 8D  84       218                 HEX   8D84
61E5: D2  C5  C1   219   READ          ASC   "READ NAME TEXT FILE"
61F8: 8D  00       220                 HEX   8D00
                   221   *
61FA: 20  0B  61   222   RDLOOP        JSR   IO ; GET NAMES FROM DISK
                   223   *
61FD: 20  0A  62   224   CLOSER        JSR   PRINT
6200: 8D  84       225                 HEX   8D84
6202: C3  CC  CF   226                 ASC   "CLOSE"
6207: 8D  00       227                 HEX   8D00
6209: 60           228   RDFIN         RTS
                   229   *
                   230   *
                   231   *
620A: 68           232   PRINT         PLA
620B: 85  06       233                 STA   PTR
620D: 68           234                 PLA
620E: 85  07       235                 STA   PTR+1
6210: A0  01       236                 LDY   #$01
6212: B1  06       237   P0            LDA   (PTR),Y
6214: F0  06       238                 BEQ   PFIN
6216: 20  ED  FD   239                 JSR   COUT
6219: C8           240                 INY
621A: D0  F6       241                 BNE   P0(ALWAYS)
                   242   *
621C: 18           243   PFIN          CLC
621D: 98           244                 TYA
621E: 65  06       245                 ADC   PTR
6220: 85  06       246                 STA   PTR
6222: A5  07       247                 LDA   PTR+1
6224: 69  00       248                 ADC   #$00
6226: 48           249                 PHA
6227: A5  06       250                 LDA   PTR
6229: 48           251                 PHA
622A: 60           252   EXIT          RTS
                   253   *
                   254   *
622B: 00           255   EOF           BRK
                   256   *
                   257   *
```

To understand how it works, consider these conditions:

Data will be stored in the area from $1000-$14FF. This area is called a *buffer*. A total of five strings will be stored, each beginning at an even page boundary like $1000, $1100, or $1200. It is assumed that no string will be longer than 255 bytes (a fairly safe assumption, since the input routine won't allow this, either).

A zero page pointer (cleverly labeled PTR) will be used to control which range in the buffer is currently being accessed for a particular string.

The basic routines used to make the overall idea work are as follows: (1) An input routine using the Monitor ($FD75 = GETLN). (2) A print routine using a JSR and a stack manipulation (not the *data* type). (3) A single-key input routine present in the Monitor used to get the command key ($FD0C = RDKEY). (4) The execution of DOS commands from machine language by preceding phrases with a control-D.

To use the program, either Brun the final code or call it directly from Basic with a Call 24576, or the Monitor with a 6000G.

A menu will appear with these choices:

1) INPUT NAMES
2) PRINT NAMES
3) SAVE NAMES
4) LOAD NAMES
5) END PROGRAM

To try the routine out, enter 1 to enter five sample names. Then use 2 to view the data you've entered. You may then use 3 to save the data as a binary file on a diskette. Then re-run the program, and verify that only the numbers 1 through 5 exist in the buffer (option 2). Then retrieve your data by using the Load command (option 4), and view again to confirm a successful load.

In detail, this is how the program works:

At entry, PTR is set to point to $1000, where the name buffer begins. The accumulator is then loaded with ASCII value for the character 1, and the CLR routine entered.

CLR puts the characters *1* through *5* into each of the string spaces. Each digit is followed by a space, and then a 0. I used zero as an end-of-string marker, but the choice is somewhat arbitrary.

MENU clears the screen and presents the user with the available choices. Points of interest here are the VTAB operation and the PRINT routine. To VTAB to a given line from machine language, one of the easiest ways is to load CV with the line you wish to go to, and then JSR to the Monitor's VTAB routine ($FC22). Normally, we might also wish either to print a carriage return or set CH to 0. Note that CV and CH are the computer's vertical and horizontal cursor position bytes, as used by the Monitor. You can always tell the cursor position by examining these bytes, and CH may be forced to a desired value to accomplish the same as an HTAB in Basic.

The PRINT routine is the one described in last month's Assembly Lines, and is useful because the JSR PRINT can be immediately followed with the data to print. This is more similar to the Basic print statement and avoids setting up a lot of specific data tables to do the printing.

Once the menu is printed on the screen, line 87 of the source file does the JSR to RDKEY. This gets the command key from the user, which is then tested by the M1 to M6 series of checks. After calling RDKEY, the keyboard value was returned in the Accumulator, and we can directly test to see which key was pressed. The key is then compared with one of the five desired responses. If no match is found, it jumps back to MENU to repeat the display and command input. Other than reset, 5 is the only way to exit the program.

Let's examine the menu choices:

If you enter 1, control is directed to the section labeled INPUT. The first thing done there is to JSR to DSPLY. Saving the detailed explanation for later, we'll simply say at this point that DSPLY just clears the screen and shows the five strings currently in memory.

After DSPLY, PTR is initialized to point to the beginning of the buffer ($1000), and the counter set to zero. The main input loop comes next. Here CTR is used to calculate what line (vertical position) to put the cursor on. (DSPLY used the same algorithm to display present data). After VTAB, the equivalent of HTAB 0 is done, followed by the jump to the actual input routine, here labeled IP. This is the same routine given last month which gets a line and then moves it to a location indicated by PTR.

There are a few subtle items in the IP routine that should be noted. The first is line 140. If return alone is entered (in other words, no new data), the routine immediately returns without rewriting the old string. This is to allow editing of a single entry by skipping the entries not of interest. Try it to see how it works.

The second item is the characteristic of this particular input routine to put the trailing zero at the end of the line. This is done on lines 141-143.

When it returns from IP, the counter is incremented and checked to see if it exceeds #$04. If not, ILOOP repeats until five strings are input. After the fifth string is entered, the program returns to the menu.

If choice 2 is entered, the DSPLY routine is called. The sole purpose of this section is to clear the screen and print the five names in memory. At entry to DSPLY, a JSR $FC58 does a HOME, and the CTR is initialized to zero. As in the INPUT section, CTR is then used to calculate the VTAB position to print each line.

D1 is the part that actually prints each line by scanning (and outputting through COUT) all the bytes at each range in-

Do You Feel That Assembler is an UPHILL BATTLE?

Let Us Help!

LISA 2.5

A 6502 Assembly Language Development System for Apple Computers. Used in homes and businesses by novices and professional programmers for over two and a half years, this most popular assembler includes these special features:

- Easy to learn commands.
- Speedy Assembly: over 20,000 lines/minute.
- Uses over 30 pseudo opcodes turning your Apple into a 16 bit machine, requiring less code to perform the desired task.
- More extended mnemonics (a greater memory aid).
- More commands which allow the flexible use of DOS.
- Runs on any 48K Apple II/II+ with or without a 16K memory expansion board.
- Only $79.95 at your local computer store.

Written by Randy Hyde, author of Assembler Programming For Your Apple, LISA 2.5 has been endorsed by Steve Wozniak, Jay Sullivan, Ken Williams, Olaf Lubeck and Mark Pelczarski.

SPEED/ASM

A set of subroutines used to simulate BASIC in developing an assembly language program. This powerful, new assembly language programming tool gives the BASIC programmer power to build complex assembly language programs and includes these special features:

- 39 pages of well written, informative documentation explain how to use SPEED/ASM and your knowledge of BASIC to write professional quality programs quickly and easily.
- SPEED/ASM is an independent Assembly language subroutine package making programs produced on your APPLE compatible with ATARI SPEED/ASM and vice versa.
- No need to own a "LISA" assembler in order to operate SPEED/ASM.
- SPEED/ASM is assembly language easily understood by the BASIC programmer.
- Runs on any 48K Apple II/II+.
- Only $39.95 at your local computer store.

LISA EDUCATIONAL SYSTEM

A complete system which offers LISA 2.5, SPEED/ASM, and Using 6502 Assembly Language, Randy Hyde's new 300 page book on assembler for beginners, all for the price of $119.95 giving you a savings of $20.00.

ON-LINE systems

Available at your local computer store from:

36575 Mudge Ranch Road Coarsegold, California 93614 209/683-6858

To order direct, send purchase price plus $1.00 for shipping (Visa, Mastercard, C.O.D. or checks accepted.)

*Apple is a registered Trademark of Apple Computer, Inc.
*Atari is a registered Trademark of Atari, Inc.

dicated by PTR. Note that as a safety check, if a zero doesn't happen to be present because of some other error, eventually the Y register will pass #$FF, and the program will fall through to D1FIN.

D1FIN provides an ending carriage return to the string and then increments CTR until all five strings have been printed.

The load/save operations are quite simple. Knowing where the buffer is located, you can access the entire block by doing either a Bload or Bsave. Remember that disk commands are done from machine language just as they would be done from Basic. The program need only output a control-D followed by a legal DOS command and a carriage return. Again the PRINT routine is used to facilitate this.

If choice 5 is entered, then an RTS is executed to end the program.

Reading and Writing Text Files. The next listing is basically a modification of the first program. If you wish, rather than retype the entire file, you can just edit the first listing to add lines 10–29, and 194–228.

The theory to the second program is fairly simple. If you think about it, the INPUT and DSPLY sections are essentially equivalent to a FOR I=1 to 5 / NEXT I type of loop that respectively Inputs and Prints five strings. In a Basic program, all that would be required to access a text file would be to precede the execution of those routines with the Open, Read and the Open, Write commands. (I'm assuming you're familiar with the normal access of Apple DOS text files; if not, read your manual!)

If you examine the new SAVE and LOAD routines you'll notice two changes. First, rather than printing Bsave or Bload, the files are opened and the read or write command output. Notice that *each* command begins with a control-D and ends with a carriage return. Second, after the command is printed, a JSR is done to the IP or DSPLY routine as is appropriate. Last of all, a Close is output before returning to the menu.

According to what we know so far, these should be the only changes needed to access text files. But there's one last catch.

Apple DOS complicates things by not allowing the user to open text files from the immediate mode. When a machine language program is running, DOS thinks we're still in the immediate mode and won't let us access the text files. What's needed is a way to fool DOS into thinking we're running a program.

This is done by using three internal management locations in the Apple. LANG ($AAB6) is what DOS uses to keep track of which language is currently running. CURLIN ($75,76) is Applesoft's register for the bytes of the line number of the program currently being executed. In the immediate mode, the high order byte ($76) defaults to #$FF. Applesoft can tell if a program is running by looking for a non-#$FF value in this location. The other way it knows a program is running is to check the location ($33) that holds the ASCII value for the prompt character. In the immediate mode of Applesoft, this is #$DD, equivalent to the] character. In a running program, this changes to #$06.

To fool DOS, all we need do is load these three locations at the beginning of the routine. Finally, when exiting the program, rather than a simple RTS, a JMP $3D0 is done to do a soft reentry to Basic. This will restore the bytes we've altered to fool DOS and return us to the current language.

Try these programs out; they make an excellent summary of many of the ideas and routines discussed so far in this series and will provide a valuable model for your own programs.

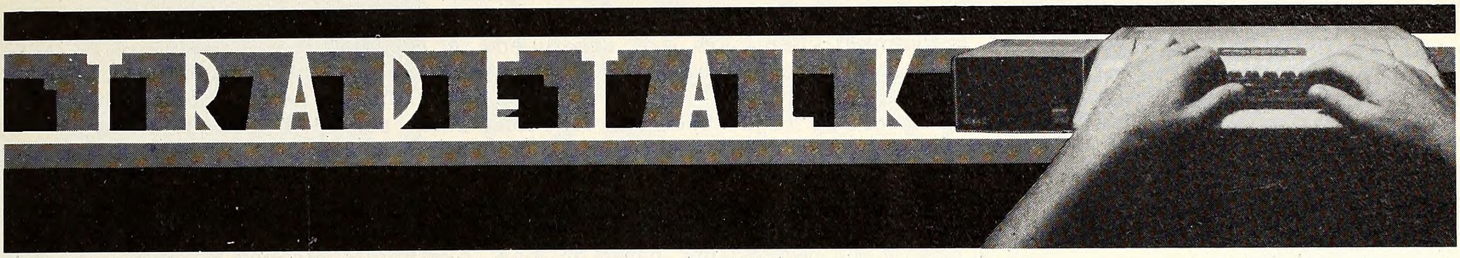
P.S. Some people have also inquired as to whether the check for a write-protect label can be defeated by modifying DOS. The answer is yes and no. Yes, the part of the code that generates the error can be eliminated, but because the write protect switch is physically wired into the recording head write system, you cannot defeat it without actually removing or altering the switch itself. Keep those cards and letters coming!

Errata. The gremlins got in the typesetting machine last month. In the program titled *Special Print Routine*, on page 66, line 26 should have read:

```
31C:  C8              26  INY
```

On page 68, lines 17 through 30 of *Input Routine for Binary* should have read:

```
                      17  *
305:  8A              18  CLEAR TXA
306:  A8              19    TAY ; T-REG = LEN NOW
307:  C8              20    INY
308:  A9  00          21    LDA #$00
30A:  91  46          22    STA (PTR),Y
30C:  B9  00  02      23  C2 LDA BUFF,Y ; PUT END-OF-STRING MARKER
30F:  29  7F          24    AND #$7F
311:  91  46          25    STA (PTR),Y
313:  88              26    DEY
314:  C0  FF          27    CPY #$FF
316:  D0  F4          28    BNE C2
                      29  *
318:  60              30  DONE RTS
```



□ **When J.E. (Jim) Howard** of La Jolla, California, bought an Apple Computer not so long ago, he found that the software programs he wanted for personal business use just didn't exist yet, so he decided to write his own. Although he was familiar with Basic and assembly language as they applied to his work as an engineer, he'd never taken a computer course or thought of himself as a programmer. But by mid-1980, he had founded a software business and begun to sell the packages he'd designed; what began as a part-time moonlight concern became a full-time business known as **Howard Software Services.** By mid-1981, he was "going crazy filling orders" for his *Tax Preparer, Real Estate Analyzer*, and other packages, so he hired his first employee early last summer; he now has five employees and plans to move the business out of his home and into a fifteen-hundred-square-foot office in December. Looking back on the remarkable growth of HSS, Howard says he wishes he'd been less hesitant about leaving his engineering career and making the firm his full-time work. "I could have been where I am now a year ago," he says.

□ Another recently founded software firm that's taken off like a rocket in the last year or so is **Synergistic Software** (Renton, WA), headed by **Bob and Ann Clardy.** Since April, they've added four new employees, including three programmers and a photo processing specialist. According to marketing manager **Will Clardy,** Synergistic keeps outgrowing office space and is already looking for its third facility since last January. Not only that—the printing firm and order processing company working with Synergistic have both had to expand their facilities, too, to keep up with those software orders.

□ **Ernie Brock,** author of the *Pascal Graphics Editor* program, has been appointed product manager at **Sirius Software** (Sacramento, CA), to be responsible for development and acquisition of new software, all prepublication tasks, and provision of technical support. Sirius has also named **Jerry Dingman** product engineer, with responsibility for product development in the hardware line as well as software, according to general manager **Jim Ackermann.**

□ A seminar for both vendors and users on "Software Quality Assurance and Configuration Management" will be held December 14 through 15 at the Copley Plaza Hotel, Boston. Primary topics will include defining the roles of software quality assurance and configuration management, as well as business structure, management tools, United States defense department requirements, and performance improvement relevant to software quality. Sponsored by *DataCommunications* magazine, the seminar will be led by **James D. Stringer, Mast Associations** president. Stringer has more than twenty-three years of data processing experience in government and industry. Registration fee, $500. For further information, contact McGraw-Hill Conference Center, New York, NY.

□ **Elaine Von Fange** has been named business manager for **California Pacific Computer** (Davis, CA), according to president **Al Remmers.** Von Fange has been with the company for one year and formerly served as administration assistant. She will set administrative policy and oversee day-to-day business operations.

□ The **California Educational Data Processing Association** will feature a seminar titled "Planning the Micro/Maxi Continuum" at its annual conference November 19 through 20 at the Capitol Plaza Holiday Inn, Sacramento, CA. **Paul A. Dali, Arthur Young and Associates,** will deliver the keynote address, "Can Micros Swallow the Main Frames?" during the first-day morning session, which begins at 9:00 a.m. Other seminar sessions will include such topics as long-range planning and integration of data processing, school administrators' role in the micro revolution, computers in special education, considerations in micro hardware selection, and implementing distributed systems. Conference fee, $90. For more details, contact Jane Householder, Office of the Los Angeles County Superintendent of Schools, Downey, California.

□ For **Waybern Corporation** (Garden Grove, CA), with more than four hundred micro dealers in its network, dealer

training is a continuing process. The network covers Arizona, New Mexico, and southern California. Waybern president **Jack Whalen** schedules an average of six meetings a year, to which all dealers are invited. Manufacturers and Waybern product line managers demonstrate equipment, discuss technical aspects, review software, present marketing plans, and answer questions.

☐ Six day-long professional development seminars will be presented immediately preceding and following the **Association for Computing Machinery**'s 1981 annual conference, to be held November 9 through 11 at the Bonaventure Hotel, Los Angeles. Three seminars are scheduled for Sunday, November 8, covering such topics as software engineering economics, graphics, and computer law. On Thursday, November 12, seminars will be conducted on security, audit, and control; software quality assurance; and introduction to software configuration movement. Seminars are scheduled from 9:00 a.m. to 5:00 p.m. each day. Registration fee, $45.

☐ According to a recent survey conducted by Dean Witter Reynolds among seventy-one computer retailers nationwide, **Integral Data Systems**' printers were ranked the major brand by 50 percent of the respondents, based on market penetration analysis. Integral Data Systems is headquartered in Natick, MA.

☐ The National TRS-80 Microcomputer Show, held last May in New York, has become the **Eighty/Apple Computer Show,** to be held April 2 through April 4, 1982, at the New York Statler Exposition Hall, New York City. Exhibitors should reserve booths now. Contact **Ken Gordon, Kengore Corporation** (Franklin Park, NJ).

☐ **JMC** (**James M. Conway**, Bensenville, IL) has formed a new division to specialize in the distribution of personal computer accessories. Emphasis is on software for small business, educational, and personal finance applications; arcade and other computer games; a variety of books on personal computers from beginner level to advanced; and other accessory items.

☐ **General Electric's Instrumentation and Communication Equipment Service Department** service center has moved to a larger facility at 1200 Kona Drive, Compton, California. Formerly located on Hindry Avenue, the center tests, repairs, maintains, and calibrates electrical and electronic equipment, including modems, computer peripherals, and telecommunications equipment. Twenty-four hour in-shop and on-site service.

☐ *Time* magazine, which holds a high journalistic tradition for reporting earth-shattering news, didn't miss the action in Coarsegold, California. **On-Line Systems** and its ad for the computer game *SoftPorn* found its way to the pages of *Time* in the midst of other world events reported in the September 24 issue. *Time*'s feature story on the versatility of computer software apparently indicates that the editors thought *SoftPorn* illustrated just how versatile it can be.

The ad, in case you missed it in *Time* and *Softalk,* shows a hot tub with three naked women in it, discreetly submerged in water. The models are **Roberta Williams** (co-author of the Mystery Adventures and On-Line president **Ken Williams's wife**), **Susan Davis,** On-Line bookkeeper, and **Diane Siegal,** production manager. Although Ken Williams griped that *Time* "hardly even printed my name," all the women involved were absolutely thrilled with the exposure. So was the town of Coarsegold, population 300, where the daily paper ran a front-page article in honor of the event; Coarsegold was proud to be singled out.

Williams says the ad has brought in only a few hate letters; most of the feedback has been positive. "People seem to like the idea of hot tubs. That's why we hold all our board meetings in hot tubs."

Two new employees at On-Line will now be attending some of those heated meetings: **Gary Kevorkian,** formerly of **Programma International** (Los Angeles, CA), and **Jeff Stephenson,** who left **Software Arts** (Cambridge, MA). Kevorkian is On-Line's new customer support manager, and Stephenson, a programming expert, has found more than enough work at On-Line. Both arrived in Coarsegold with no prior experience in hot tubs.

However, Ken's brother **John Williams,** who handles the company's advertising and has evidently gotten into enough hot water, is leaving to open his own business: J and J Marketing. (The other "J" belongs to Jill Lennen.) The new company will be handling the advertising for both *On-Line* and *California Software.* ■

Introducing The Grappler™:

The only interface that makes computer graphics easy as Apple™ pie.

ACTUAL APPLE II PRINTOUT USING GRAPPLER™ AND EPSON MX-100.

Orange Micro makes the only universal parallel interface card that simplifies high resolution graphics for Apple™ computers. No longer does the user need to load clumsy software routines to dump screen graphics — it's all done by The Grappler's™ exclusive E-PROM chip. There are versions to accommodate the Anadex, Epson MX-100, MX-80* and MX-70, IDS Paper Tigers, Centronics 739, and *NEC* Prowriter, and future graphics printers. The Grappler™ accepts 18 simple software commands accessible through the keyboard or user program, making it the most intelligent Apple Interface available.

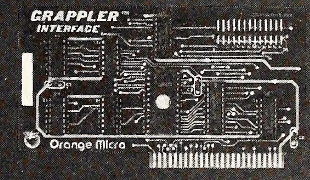
Order The Grappler™ through Orange Micro direct, or authorized dealers.

*Requires Grafftrax-80

FEATURES — User benefits

SELECT PAGE—Choice of Hi-Resolution Graphics page 1 or page 2.
INVERSE GRAPHICS—Provides reverse graphics of black-on-white or white-on-black.
DOUBLE SIZE PICTURE—Doubles the graphic screen representation vertically & horizontally.
90° ROTATION—Rotates the screen picture 90°
CENTER GRAPHICS—Accomplished through setting left margin thereby centering the graph.
CHART RECORDER MODE—Successive horizontal pictures are combined continuously simulating a chart recorder.
BLOCK GRAPHICS—For printers with block graphics (e.g. Epson MX-80, Okidata M80), the high order bit can be controlled.
BELL—For printers with a bell, bell characters are deleted during user program listings.
MARGINS—Set left and right margins.
SKIP-OVER-PERF—Set page length; printer will automatically skip 6 lines between each page.
VARIABLE LINE LENGTH—For user program listings, sets line length and wraps around with breakpoint at nearest blank.
TEXT SCREEN DUMP—The text from a user report or page of program listing can be dumped directly from the screen.
WORKS WITH PASCAL & CPM

The Grappler™

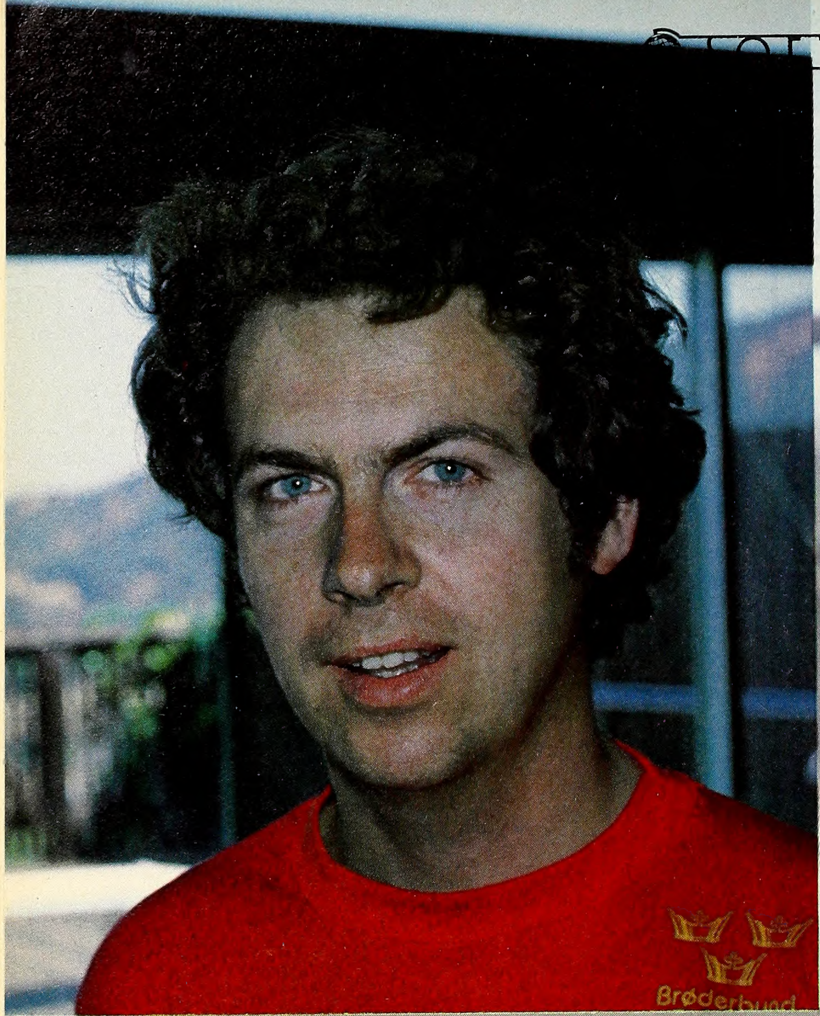


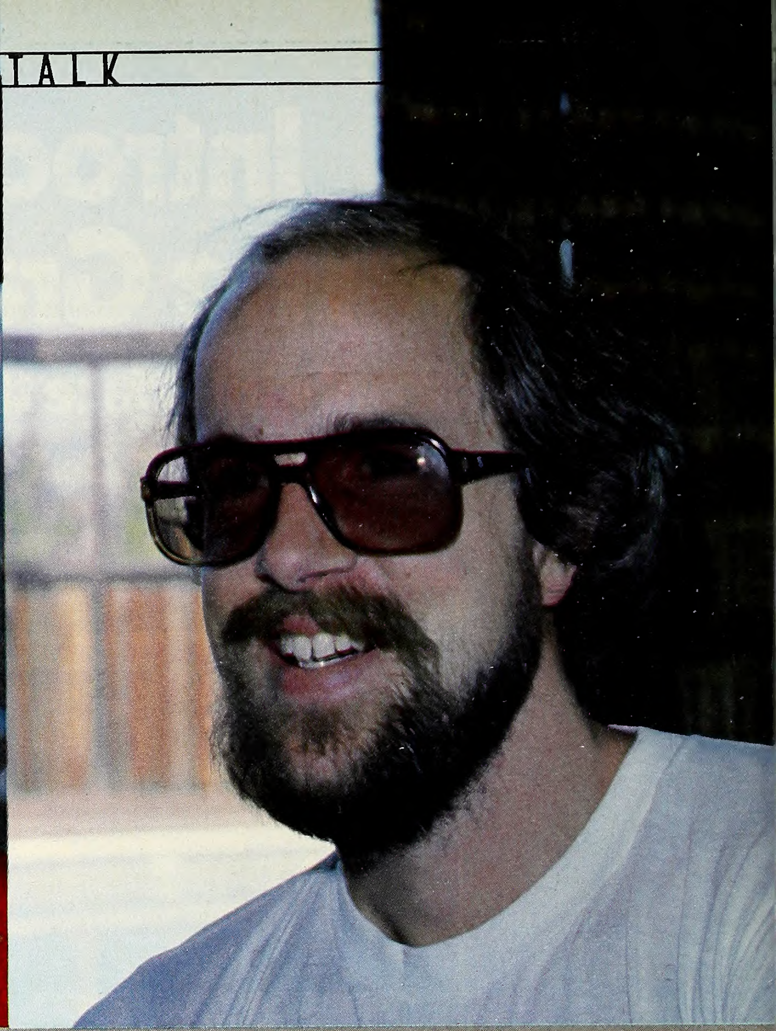
ANAHEIM STORE
3150 E. La Palma, Suite I
Anaheim, CA 92806
(714) 630-3622

SHERMAN OAKS STORE
13604 Ventura Blvd.
Sherman Oaks, CA 91423
(213) 501-3486

Orange Micro inc.
3150 E. La Palma, Suite G, Anaheim, CA 92806
(800) 854-8275 TOLL FREE
(714) 630-3322 CA, AK, HI

Dealer inquiries welcome to:
Waybern Corporation
(714) 554-4520
CompuCable Corporation
(714) 635-7330





Exec Brøder

Saga and Star Craft Spell Success

Above, the Carlstons: Doug, Gary, and Cathy. Below, programmer Chris Jochumson, production manager Brian Eheler, who accompanied Broderbund from Oregon, and Minoru Nakazawa, president of Star Craft of Japan.

BY ALLAN TOMMERVIK

One of the salient features that make the microcomputer industry such a pleasure is that most of the people who populate it are genuinely fine individuals.

This is not to say that one cannot find wonderful human beings among the populations of garment workers, automobile mechanics, insurance brokers, or head hunters. But any knowledgeable observer would concede that the incidence of truly thoughtful, considerate people is higher in the software publishing industry than in any other calling except perhaps the ministry.

Certainly nowhere in the industry are there Boss Tweeds or Machiavellis brokering power. Neither are there Borgias dispensing with their rivals permanently. Even lacking for the most part are the pettinesses and jealousies that accompany most rivalries even outside the business world.

No Buts About It. In most businesses, the comment that someone is a nice person is a left-handed compliment, almost always followed by "but . . ." and a long accounting of his limitations, real or imagined. Praise of an individual's character is almost universally damning in the normal business environment.

But descriptions of people working in the Apple world almost always include testimonials as to character and hardly ever carry any qualifiers as to talent.

There seems to be no cosmic sociological truth that would account for the attraction of a good people to this industry—certainly data processing in the mainframe sense has only its share of exceptional people and no more.

Yet a roll call of the men and women currently working in Appledom would turn up considerably more people who enhance the stature of the human species than any Vegas bookie would consider likely.

In fact, close scrutiny of the industry almost reveals an ongoing morality play in which the unscrupulous and the hustlers are weeded out and the good folks, tried and true, succeed beyond their expectations.

This is not to indicate the the microcomputer industry is populated with late twentieth century Jack Armstrongs. The age of innocence and naivete triumphing has passed. But nowhere is there an industry where so many of the principal players can be trusted to keep their word on a handshake deal.

These observations would have pertinence to almost any prior article in the Exec series. But they're most appropriate to accompany this profile of the people at Broderbund because, far more than any other successful software publisher today, the Carlstons of Broderbund have succeeded on the strength and virtue of their character.

The Law on Programming. Almost every other successful software publisher built from the foundation provided by one solid programmer. Personal Software succeeded on the strength of partner Peter Jennings's *Microchess*.

On-Line Systems had Ken Williams, Sirius Software had Nasir, California Pacific had Bill Budge, Southwestern Data Systems had Roger Wagner, Stoneware had Barney Stone, and Synergistic had Bob Clardy.

What Broderbund had was Doug Carlston. In the battle of peeks and pokes, JMPs and JSRs, hi-res graphics and animation, Broderbund came armed with a few Gotos and Gosubs. Talk about your long odds, Doug Carlston was Broderbund's premier programmer, and he not only didn't have an Apple—he started on a TRS-80—he was an attorney.

Mark Twain once defined an attorney as the man with his hands in someone else's pockets. Attorneys are the guys looking for those plush judgeships; they're not accustomed to any more adversity than losing a case taken on a contingency fee and expenses. And they certainly are not known for being conversant with binary and hexadecimal number systems. Any number system that's not functional for measuring fees and tallying golf scores is definitely foreign to your basic legal eagle.

From Red Sea to Lake Michigan via the Charles River.

Broderbunders at work overlooking San Rafael. Back row from the left: Brian Eheler, Cathy Carlston, Gary Carlston, Chris Jochumson, and Doug Carlston. Front row from left: John McWhorter, Marcia Costiglione, Kim Brown, and Michelle Keller.

Doug Carlston defied all these stereotypes. He took a leave of absence from Harvard after his sophomore year to work in Africa. As part of a volunteer Harvard organization, he taught mathematics and geography. That same high-minded volunteer group, sans Doug's help, erected the largest potato chip factory in Africa, enabling the Ethiopians to eschew their natural diet in favor of junk food.

Having been certified by Harvard's dons as having sufficiently deft hands to place in other folks' pockets, Doug joined one of the highest—if not one of the largest—law firms in North America, their offices being on the eighty-second and eighty-third floors of the Sears Tower in Chicago.

For two years he labored there, working on the knotty problem of Lake Michigan water rights. The waters of this lake are of exceptionally high quality, with the result that every governmental body within driving distance bids to get the water for their constituents. The requests for water were several million acre feet in excess of supply, causing several appointed Solomons, Doug among them, to toil for years determining who should have rightful claim to how much of that water.

Doug and Life among the Lobsters. If this sounds reasonably mundane, perhaps even boring, you've been far too isolated from the real world where the basic necessities of life are contemplated. There is no more controversial area of American society than the question of water rights, and attorneys skilled in adjudicating the various claims have their futures well ensured.

So Doug Carlston threw over what appeared to be a certain future when he took up what he terms "boondock" law, meaning that he went to a small rural area and set up private practice.

This small rural area happened to be in Maine, and to grasp the concept of small as it pertains to this and similar spots throughout rural America you should understand that the headstones in the cemetery outnumber the living inhabitants.

It is an article of faith that population centers consisting of all of two hundred people are notoriously lacking in intellectually stimulating social events, such as plays, concerts, and the like.

So it came to pass that Doug Carlston hied himself off to the big city and purchased a TRS-80 with which to while away the hours.

Gary and the Swedish National Champion Women's Basketball Team. Meanwhile, younger brother Gary, also a former inmate of Harvard, was following an even more obscure path to a software publishing career.

Gary Carlston matriculated to the Cambridge site with the idea of studying architecture but found that not a possibility. Lacking a further definition of a desirable course of study, he opened the subject to several of his classmates, who recommended an arcane sociological subject matter that was also unavailable to him.

But in pursuing this line of action, he came across the Scandinavian studies section. Being Swedish by heritage, and having access to a department head after several terms of being taught by teaching assistants and under profs, this academic niche seemed a natural.

After graduation, he traveled five times to Sweden, staying a total of five years. During that time he coached the women's basketball team that became the Swedish national champions.

His return to the States found him teaching Swedish at the University of Washington, after which he took a position in Eugene, Oregon, as director of the March of Dimes for that state.

Almost an Entrepreneur. It's Gary Carlston who had the nascent entrepreneurial urge in this family of academics—their father is a professor of new testament studies and their brother is a professor of social psychology.

But Gary's early attempts at business success always managed to fall short. With a partner, he imported safety reflector materials to be worn by schoolchildren. The partnership had a modicum of success getting individual retail stores to handle the product, but they were unable to find a single national distributor who felt that decorating children like bicycles was up their alley.

Gary's second business, in conjunction with Doug, actually became a big success, but not for them. Doug conceived the idea of conducting tours for Europeans and Japanese in the

United States. Gary lined up common carriers, tied down tour and lodging reservations and was about ready to begin taking in tickets when they were stymied by the fact that United States citizens must get approval from the Interstate Commerce Commission before commencing such activities. All their money had gone into the other preparations, so they couldn't afford the cost of applying to the ICC for permission.

A British company, not needing ICC approval because they weren't United States nationals, picked up the advertising concept and itinerary and made the venture a significant success.

Cathy and the World of High Fashion. In the meantime, sister Cathy Carlston had graduated from the University of Massachusetts and headed for Gotham. She got a job at Lord and Taylor, where she worked herself up to being a buyer.

Unlike her brothers, who started at or near the top and worked their way down—Doug from a big-time law firm to a small Maine practice and Gary from coaching the Swedish national women's basketball champions to peddling safety reflector material—Cathy had followed the traditional route and achieved early a most prestigious position.

Lord and Taylor is not Nieman Marcus, but neither is it your local ready-to-wear store. And the position of buyer for an influential New York store carries with it a certain ability to affect styles and fashions throughout the country.

So in late 1979, prior to the formation of Broderbund, Doug was ensconced in Maine toying with his TRS-80, Gary was hyping walkathons for the Oregon March of Dimes Foundation, and Cathy was in the mainstream of New York fashion.

But Doug was taking the steps that would change all of this on a permanent basis. He was developing the *Galactic Saga*, originally a trilogy of strategy games with a political and philosophical base.

The Extraordinary Effects of Financial Success. When he had the programs under reasonable control, he submitted copies to Scott Adams at Adventure International and to TSE Softside. A short time later, he received a $300 royalty check from Adventure International. With that as an impetus, he renewed importuning brother Gary to join him in forming a software publishing company.

Gary had been firm in rejecting that idea earlier, his past business experiences not lending credence to the thought that entrepreneurship could be lucrative. But in the face of Doug's success at actually making money on software, his resistance crumbled.

Maine to Oregon, or vice versa, being a tad too far to commute on a daily basis, it became essential that one of the brothers relocate. It was Doug; he decided to join Gary in Eugene.

All was not tea and crumpets for the Carlstons forevermore, however. Those early TRS-80 programs were rife with bugs. Doug was making almost daily updates to the master disks to clean up the programs, but the various publishers were slow to switch from already packaged product to the newer versions.

From that experience came the determination never again to release software before it had been completely checked out. Writing and publishing bug-free software is no piece of cake as Doug Carlston can attest—the fourth episode in the *Galactic Saga* series was six months late hitting the marketplace.

The brothers were in full gear in time for the 1980 West Coast Computer Faire. They took a microbooth to display their three TRS-80 programs in the *Saga* series, as well as a lo-res *Tank Command* written by their brother Don.

The Computer Faire has been a launching pad for several companies; Apple itself can trace its first successful marketing steps to this show. But for all the various values that the Faire provides for both consumer and businessman, Doug Carlston believes that the microbooth may be the most important innovation of show sponsor Jim Warren.

The microbooth provides inexpensive access to a large number of end users, giving a fledgling company a chance to test the marketplace acceptability of its products.

Microbooth Led to the Stars. Having a presence at the Computer Faire also validates a company in the eyes of the rest of the trade, and it's here that participation in the Faire in 1980 made the biggest contribution to the future of Broderbund.

It was at that Faire that Broderbund made the Star Craft connection that would eventually vault them into the first line of software publishers.

Star Craft is a Japanese firm headed by Minoru Nakazawa that includes custom programming for larger systems among its business activities. The American microcomputers had made some inroads in Japan, and Nakazawa sensed that the micros presented additional programming opportunities.

There were not sufficient numbers of the machines in Japan to warrant any full-fledged business effort, so Nakazawa turned his attention to the American market, using the concentration of software publishing companies at the Faire to simplify his screening process.

Sold on Company, Product Unseen. Although he was unable to see samples of Broderbund product at the Faire, there was something about the Carlstons that prompted Nakazawa to pursue further a business relationship.

That something that Nakazawa saw was character, which brings us full circle to the beginning of this article.

Nakazawa's original impression was reinforced by viewing Doug's *Galactic Saga* series. Although that series experienced modest success in the American marketplace, it was never a success of top bestseller proportions.

But Nakazawa saw in it elements that apparently touch the Japanese psyche more directly. His feeling was roughly analogous to reading a book and knowing that the author is a person you would enjoy and value knowing.

Nakazawa's perception has been ratified by the greeting *Galactic Saga* has received in Japan, where the programs rival Bill Budge's packages as the most popular items for the Apple.

So it was that the Star Craft-Broderbund connection came

Two classic board games that continue after half a century to bring people together for fun.

Now you can play these board games with your Apple personal computer!

MONTY® Plays Monopoly® is a computer opponent for America's most popular board game. MONTY Plays Monopoly rolls the dice, keeps track of each player's position and collects $200 when passing Go. Up to seven players, plus MONTY, can play at one time. Available on diskette for the Apple II with 32k, one 13-sector disk drive, a Language card or AppleSoft card, and game paddles, or the Apple II Plus with 32k, one 13-sector disk drive and game paddles.

Your Apple can also join you in the Scrabble® brand crossword game, the world's best-selling word game.

MONTY® Plays Scrabble® has a vocabulary of over 54,000 words and plays at four levels of difficulty. Up to three people can play with MONTY, who will keep score for everyone. Available on diskette for the Apple II with 48k, one 16-sector disk drive and a Language card or AppleSoft card, or the II Plus with 48k and 16-sector disk drive.

Both programs act as computerized opponents rather than being computer reproductions of the games themselves. The complete Monopoly and Scrabble sets are required.

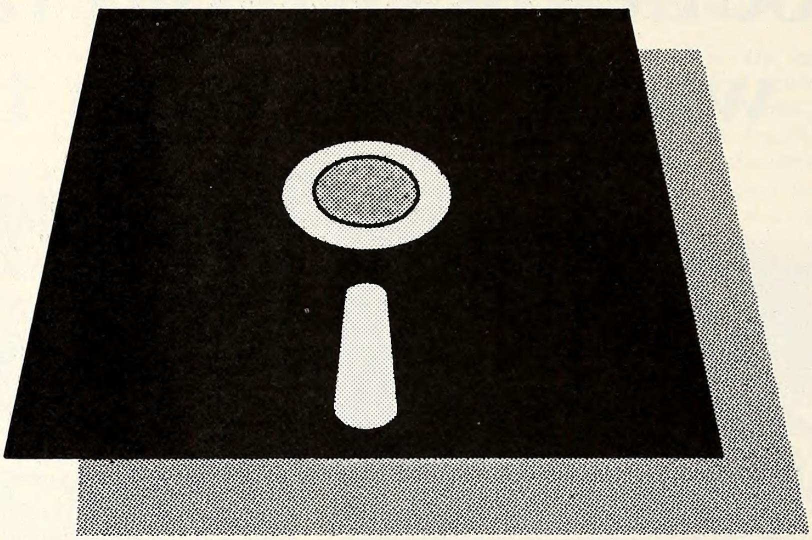
Both computer programs were created by Ritam Corporation. See them at your local retail computer store.

MONTY is a trademark of Ritam Corporation; Monopoly is a registered trademark of Parker Bros., which does not sponsor or endorse MONTY Plays Monopoly; Scrabble is a registered trademark of Selchow & Righter Company for a line of word and sentence games, and it does not sponsor or endorse MONTY Plays Scrabble; Apple and Applesoft are trademarks of Apple Computer, Inc.

Introducing two old favorites that won't fade with the fads

$600

All You Ever Need To Spend



Jb

TITLE: "THE PROGRAM WRITER/REPORTER* "

Enables ANYONE to write complete, running, debugged BASIC LANGUAGE Programs in 35 to 40 minutes with NO PRIOR PROGRAMMING KNOWLEDGE OR ABILITY.

IF you are one of the many who bought a microcomputer in the belief that with just a little studying you could write your own programs, you now know that you can't.

IF you, as a businesman, thought you could have stock software modified at a reasonable cost with reasonable results, you know that's not possible either.

IF you are a hobbyist getting tired of the untold hours it takes to write a program, only to find it takes more hours to debug than to write.

IF you are a skilled programmer you don't have to be reminded of the repetitious time spent on each new application.

IF you have left your micro-computer sitting somewhere gathering dust . . . meet THE PROGRAM WRITER/REPORTER*.

THE PROGRAM WRITER/REPORTER* is not just another data base generator.

THE PROGRAM WRITER/REPORTER*, at your direction, makes complete running programs that are thoroughly documented, easy to modify at any time by YOU!

THE PROGRAM WRITER/REPORTER* cuts programming time up to 90% for a skilled programmer.

THE PROGRAM WRITER/REPORTER* will make anyone a skilled programmer in 30 to 35 minutes!

THE PROGRAM WRITER/REPORTER* does the work! You can answer the simple direct questions and THE PROGRAM WRITER/REPORTER* CREATES . . . AND ALL IN BASIC LANGUAGE.

Q. After THE PROGRAM WRITER/REPORTER* has produced a program, can it be modified?
A. Yes, the resulting program is modular, fully documented and readily accessible for alterations or deletions.
Q. Does the program created use so much disc space that there is very little space left for the record storage?
A. No, the code produced is extremely compact despite complete documentation. If requested THE PROGRAM WRITER/REPORTER* will even "pack" or compress information. You may even delete the "remarks" making it even more space efficient.
Q. Must I be expert or even conversant with Basic Language?
A. No, all questions to and answers from the operator require no computer language knowledge, simple every day English will do.
Q. What about math ability?
A. If you can count your fingers and toes, you'll have no problems.
Q. Will the programs which I produce with THE PROGRAM WRITER/REPORTER* be bulky, slow or amateurish?
A. No, the resulting programs will be sophisticated and extremely fast operating. For example, should you create a mailing list or inventory program, the time for any record to be retrieved and displayed from a full disc would take a maximum of 1 second.
Q. Must the programs produced conform to a pre-determined format and file length?
A. No, you determine format and file size to fit your requirements.

Q. Can I develop my own business programs?
A. yes.
Q. What are the limitations? What programs can I produce with THE PROGRAM WRITER/REPORTER* ?
A. Your own ingenuity and hardware limitations. 100's of different programs.

TECHNICAL ASPECTS

The Reporter Package - makes reports your way.
Record access by a hashing algorithm guaranteeing fast record retrieval.
Duplicate keys permitted.
Record deletion automatically supported.
Record access and file maintenance is user transparent.
Minimal disc overhead since there is no special assembly language routine called. No "Basic" overhead.
Programs produced can be transported between 6800, 6502, 8080, Z80, 8085, 8086 and Z8000 based systems.
Can be used with Micro-Soft Basic and CP/M systems.
Complete file maintenance including up-date of any record in any field, delete and add new records even with duplicate key.

VANLOVES

800-255-5119

Vital Information Inc. 7899 Mastin Dr. Overland Park KS 66204

to pass, not because Doug Carlston was a better programmer than Nasir or Ken Williams, and not because the Carlstons were better businessmen than Dan Fylstra of Personal Software, but because Nakazawa perceived in them qualities of character that he valued in people with whom he partnered.

The *Softalk* Bestseller columns have since recounted how Star Craft's *Apple Galaxian*, now called *Alien Rain*, surpassed *VisiCalc* at Christmas last year to become the hottest selling software for the Apple, breaking what had previously looked like *VisiCalc*'s permanent possession of the top spot.

Snoggle, *Alien Typhoon*, and *Apple Panic* followed, each finding bestselling acceptance among Apple owners.

But it wasn't all that simple.

A Taste of Lean Times. Their first Computer Faire was in April 1980, and by July of that year the Carlston brothers had exhausted the seed money they had extracted from various relatives and prospects looked dim.

The *Galactic Saga* had been translated to the Apple, but Broderbund's phone was not exactly overtaxed handling orders. At the behest of friends, Gary visited in the Bay Area for a weekend; and while there he managed the first sales that generated some impetus for the company. July became the first month for which the company showed a profit.

The Carlstons credit the support of several retail stores for getting them through this lean period. Included among them are Computerland of Nashua, New Hampshire; Computerland of Cleveland, Ohio; The Program Store, Washington, D.C.; Data Domain of Schaumburg, Illinois; Computerland of Downers Grove, Illinois; and Computerland of South King County, Washington.

During this time, they were handling their own sales because they had not yet reasoned out a pricing policy that could include distributors. One reason for their inability to give distributors a price break was that they were selling some of their Star Craft product at a loss.

It wasn't intended that way, but by the time they paid duty on finished product coming into the United States, the duty plus the payment to Star Craft exceeded by one dollar the price they were charging the retail stores. Distributors showed no desire to share that negative margin.

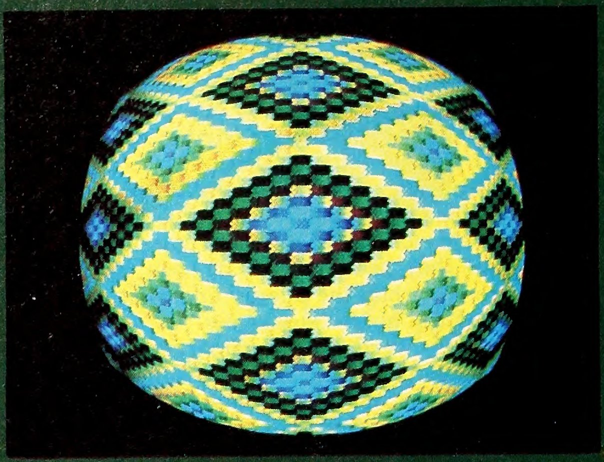
Nevertheless, Broderbund was making steady progress that augured well. Then came *Apple Galaxian*. They showed the program in prerelease form at a Boston computer show in November and knew they had a big hit on their hands—even though the advance copy had the enemy dropping umbrellas instead of missiles at the player's cannons.

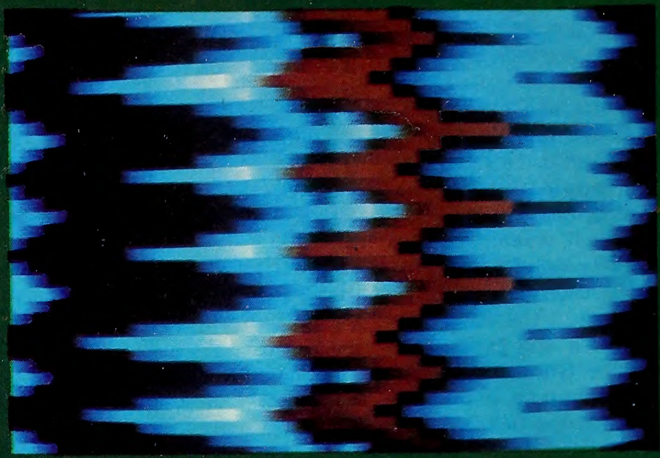
Simultaneously, they were able to construct a distributor policy that attracted Softsel. With Softsel's support, Broderbund got off a demo copy of Apple Galaxian to every retail store, starting a deluge of orders that caused the company's volume to nearly sextuple in one month.

Character Has Its Own Rewards. Broderbund's operation gained in profitability when Nakazawa decided to entrust the disk masters of Star Craft's programs to the Carlstons. This show of faith enabled the company to avoid the per piece duty on product by manufacturing the finished software in the United States.

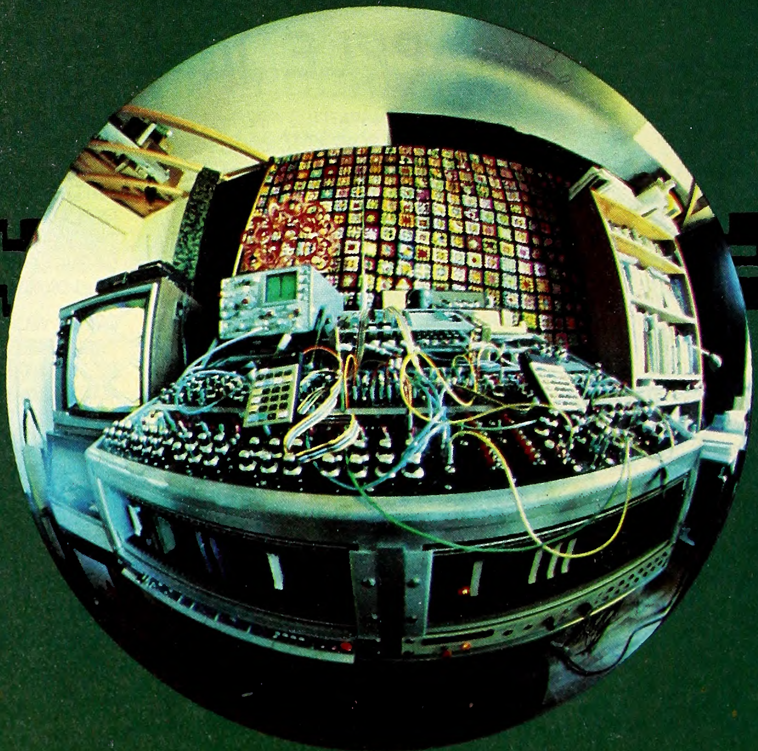
The addition of Star Craft's *Snoggle* gave Broderbund a powerful one-two punch in the Apple marketplace, and it

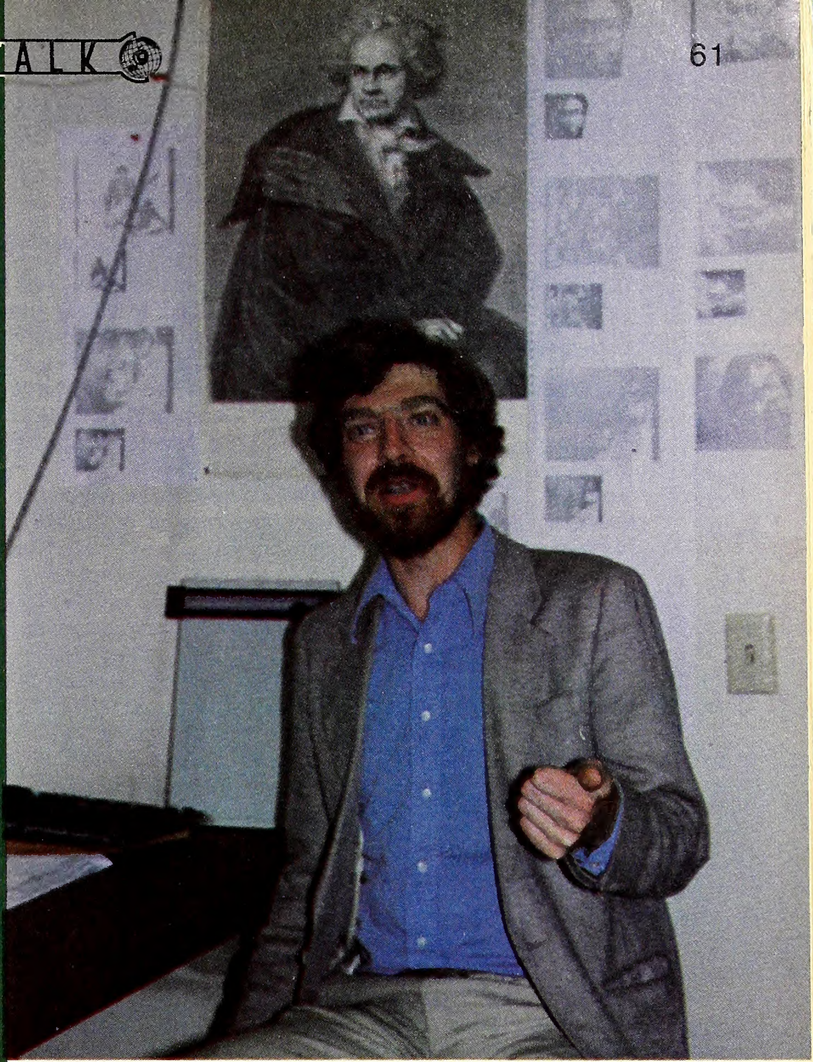
GOTO 74





Steve Beck:
Artist of the Future





Above left: two examples of art produced on the Video Weaver. Above: a frame created on the Direct Video Synthesizer. Right: a portrait of the artist and friend. Immediately below: Stephen Beck at the Direct Video Synthesizer and the Video Weaver. The smaller object above Beck's right hand is the Video Weaver. Farther below: a wall full of energy controllers designed by Beck-Tech. Below left: another look at the Direct Video Syntheizer.

BY CRAIG STINSON

The offices of Beck-Tech, in Berkeley, California, present a scene of subtle contrasts. The locale is the Claremont Hotel, a massive, sprawling piece of early twentieth-century grandeur that evokes thoughts of Warren Harding and Silent Cal Coolidge. The Beck-Tech suite, tucked into a corner of this venerable beauty, is filled with computers and other modern electronic gear.

Pictures of other worlds adorn various walls of the office. There are satellite photos of Mars and Jupiter and images of the earth from nonterrestrial vantage points. These space photos awaken in the visitor a respect for beauty and changeless realities and a conviction that the occupants place a high value on long-range vision. Around the corner from Mars and Jupiter, a poster-sized portrait of Beethoven surveys a room full of Apples and paraphernalia, bestowing a sense of range and vision of another kind.

Clearly, Beck-Tech is a place of high purposes and serious work. Yet one also sees here a shelf full of electronic toys—a Star Wars game, a hand-held horoscoper, and other frivolities of the microcomputer age.

A Man of Many Talents. All this diversity reflects the character of Beck-Tech's proprietor, who is himself a man of striking appositions. Stephen Beck is, among other things, electrical engineer, artist, musician, and inventor.

His principal artistic medium is video, a form of expression that he has, on occasion, described as visual music, to stress—not its interaction of sight and sound—but, rather, its kinetic nature. Like music and unlike painting, Beck's art takes place across time; it moves from beginning, through middle, to end.

Beck has been involved in the field of video art since the late sixties. His works, for several of which he has written original music, are on display in such places as the Museum of

Modern Art and the Whitney Museum of American Art in New York City, the Museum of Contemporary Art in Caracas, Venezuela, and the Musee d'Art Moderne in Paris. Many of his compositions have been aired on PBS television.

Beck is one of a handful of artists working in the area of nonrepresentational video. His artistic heroes, he says, have been painter Wassily Kandinsky and the German film maker Oskar Fischinger; the latter was the first to develop a nonobjective kinetic art through animation.

"Personally," Beck says, "I have been interested in the symbolic, ideographic, and nonobjective modes of images, those which originate internally within the mind's eye." To this end, he sought to make television a means of re-creating, among other things, hypnagogic and hypnopompic imagery—the kinds of pictures we see just before we fall asleep or during the process of awakening. Phosphenes—those colored flashes of light that dance before our eyes when we rub closed eyelids—have been another source of artistic material.

Man Cannot Live on Art Alone. But if this predilection for the nonobjective lends a sort of right-brain bias to his art, it is certainly balanced by other aspects of his professional life, for Beck is a practical, problem-solving person—an artisan as much as an artist.

The principal activity of Beck-Tech is the development of hardware and software tools for the field of energy management. Products the company designs now conserve money, energy, and manpower for large retail chains all around the country.

In his artwork as well, Beck has been a toolmaker. He has invented two generations of video synthesizers and a digital video weaver. The video weaver has led to the development of software that will make the Apple a tool for video weaving. Beck's video-weaving software for the Apple is scheduled to be available to the public in January.

Work is also under way to produce a peripheral card that will make the Apple's video signal conform to the government's standards for broadcast quality. This card—to be called the Chromatron—will also modify the Apple so that it will be able to produce 4,096 distinct colors.

Beck traces his interest in electronics back to his seventh birthday, when his father gave him a crystal radio set. He used to tune in Cleveland stations at night on his crystal set from his home in Chicago; and electronics, along with painting and music, became a consuming hobby.

Painting with Music. In 1968, while he was working toward a degree in electrical engineering at the University of Illinois, music synthesizers were just coming into vogue; their example led him to produce a video synthesizer—an instrument that would generate video signals directly, without the use of a camera.

At that time, the University of Illinois had a contemporary music ensemble of national prominence. Beck used to attend some of their concerts with video synthesizer in hand, making visual music to complement the ensemble's performance.

He was also in those days trying out neon as a visual medium. "I was always more interested in emitted light than reflected light," he says. "I built a kinetic tombstone with twenty-seven pieces of neon tubing. It had three ruby-red hearts that blinked in sequence, so the piece appeared to be throbbing." Beck also wrote a complex computer program to control this sculpture.

Beck's first video synthesizer was an analog device, some-

what smaller than a bread box. It still resides in his office in Berkeley.

Two years later, he produced a much more powerful second version of the synthesizer. At this time, in 1970, Beck was an artist in residence at the National Center for Experiments in Television, working under a grant provided by the National Endowment for the Arts. The National Center for Experiments in Television, in San Francisco, was the official research vessel for public television, and during his tenure there, Beck broadcast several hours of his work over the Public Broadcasting System network.

In 1973, Beck received a grant from the American Film Institute to do experimental nonobjective composition using a combination of video synthesis and film techniques. About this time, he became interested in the nonobjective imagery produced by native American weavers and quilt makers, and his career took off in an entirely new direction.

"I was looking for a way to connect my art with the traditional," he says. "I had grown tired of being avant-garde."

Back to the Loom. His new fascination led ultimately to the invention in 1974 of a video weaver—a digital device that worked in conjunction with his direct video synthesizer to produce real-time control of mosaic imagery on the CRT screen. In the case of the video weaver, frustration was the mother of invention.

"My girlfriend had given me a standard loom. I worked with this thing for about three months, trying to get it going. I finally got the warps and wefts up, and I decided this was ridiculous. I was never going to get anywhere doing this."

So Beck went back to what he knew best and adapted electronics and computer technology to an old art form. "Any computer-driven raster is the same kind of matrix as woven textile," Beck says. "I didn't try to copy their patterns, but I plugged algorithms into the video weaver and, lo and behold, what came out was like Navajo weaving." For the video weaver, Beck designed a microprocessor chip that produced pixels—picture elements—directly rather than by way of numbers.

In 1976, Beck licensed one aspect of his microprocessor to National Semiconductor and began fifteen months of consultation with that company on the development of a television game chip set.

Work on microprocessor chips for game applications eventually led to an association with Eddie Goldfarb, of Los Angeles, who had been making games of all sorts since the 1940s. Beck and Goldfarb together were among the first to get into the business of making hand-held electronic games built around small dedicated microprocessors.

The two produced a series of games over the next several years, including a musical version of Concentration called Melody Madness and a hand-held electronic astrology calculator that used a 2K microprocessor to produce detailed horoscopes.

It was about this time that the Apple computer appeared in the marketplace, and Beck adopted the new machine as a sketch computer for the development of game chips.

Hopping with Energy. Beck-Tech became a corporation in 1978 as a hardware and software consulting firm. The company started out writing single-chip programs and, through the intermediary of Texas Instruments, came in contact with another company, Margaux Controls of Santa Clara, California, which was working in the field of energy management systems.

Beck-Tech's first product for Margaux was a single-chip Timemaster that allowed for the programming of eight circuits for time control. Subsequent work led to the development of complete systems for monitoring temperature, pressure, and other energy parameters, using the Apple as a communications device.

To date, the Beck-Tech–Margaux collaboration has found its principal application in large retail store chains like Safeway and Penney's. Traditionally, the function of energy monitoring and management had been carried out by mechanical devices, without any kind of centralized control. Under the system developed by Beck-Tech and Margaux, a regional supervisor can dial up individual stores on his Apple, over standard voice-grade phone lines, and monitor or alter factors that affect energy consumption.

Besides making the process of energy management simpler and more flexible, the system has averted a few disasters—as in the case of a Safeway in Houston where a mechanical thermostat in a meat locker jammed one night. The rising temperature was sensed by a Beck-Tech monitoring unit, which then set off an alarm in a central office, allowing the meat to be saved.

A Visual Concert. The newest child of Beck-Tech is a group called Electron, consisting of Beck, Henry Spragens, and other players to be announced, which is putting together live musical/video performances using Apples. Their first concert is scheduled for Berkeley's Claremont Hotel late this month, and they have been invited to perform at the 1982 Video Conference of the American Film Institute in Los Angeles and Washington, D.C. next June.

Beck has also recently finished making a segment for a television series called "Computer Chronicles." On his segment of the show, Beck will explain his video-weaving algorithms. Produced by KCSM in San Francisco, the series is scheduled to begin airing over the Public Broadcasting System in November. ■

VENTURES WITH VISICALC

BY CRAIG STINSON

Last month this column reported a bug in the sixteen-sector version of *VisiCalc*. The problem—acknowledged by Personal Software—occurs when more than seven DIF files are saved on the same disk. Occasionally under such conditions the disk catalog crashes, making some or all of the files inaccessible.

Personal is in the process of making amends for this error (see this column, October 1981, for details). In the meantime, however, we have a report from a reader—who prefers to be anonymous—that the problem occurs only when the arrow keys are used to step through the disk catalog during a save or load operation. If you don't mind typing in the file name, instead of pulling it out of the directory by means of the arrow keys, then—according to this informant—you won't run the risk of losing your data.

We've not been able to confirm this information, but we can say that we haven't been able yet to crash a disk without using the arrow keys. So this tip comes your way for what it's worth. If you use DIF files and you're waiting for an update from Personal Software, or if you just don't want to be bothered with getting an update, your data may be safer if you get your typing fingers up and lay off those arrow keys.

Meanwhile, Saturn Systems (Dearborn, MI) is offering relief for a different kind of problem associated with sixteen-sector *VisiCalc*—reduced memory space.

The DOS 3.2 version of *VisiCalc*, on a 48K Apple, left 25K of user work space. The sixteen-sector version cut that area by about 25 percent—down to 18K. To a small extent, the newer *VisiCalc*'s DIF feature may have compensated for the memory reduction, since it made it easier for the user to break up large models and reconsolidate components as needed. Nevertheless, 7K was a fairly steep price to pay for the enhancements of the new *VisiCalc*.

Fortunately, the 3.3 *VisiCalc* was designed to take advantage of the Language System or any other 16K RAM card. If you have such a device, your work area is expanded to 34K—a comfortable enough space for many applications, if not for all.

For those with large-scale *VisiCalc* applications who need still more room to ramble in, Saturn Systems has a solution. The company makes a 32K RAM board (reviewed separately in this issue, page 84). They also offer an optional piece of software—for $100—that will allow *VisiCalc* to make full use of all that on-board RAM. You can, in fact, use the *VC Expand* software to exploit any RAM card or combination of cards.

With a single 32K board and no language system, your useful space expands to 50K. With a second 32K board you can get 82K—a little more than what you'd have on a minimally configured Apple III.

If you had nothing but numbers on your worksheet, an increment of 32K would be worth about 3100 additional cells. If you had a lot of formulas as well, that 3100 figure would drop off considerably—depending on the complexity of your formulas.

There is one problem with the *VC Expand* software that remains to be solved. If you have the Language System as well as another RAM board, the Saturn software will recognize both. If, for example, you have the Language System and one 32K Saturn board, you'll get a total of 66K of user space. But 2K of that space will not be reliable because of the presence of the autostart ROM on the Language System board. The autostart ROM rules the territory from $F800 to $FFFF. A Saturn spokesman said the company will probably cope with this annoyance with a software change giving users the option of cordoning off the 2K block of memory.

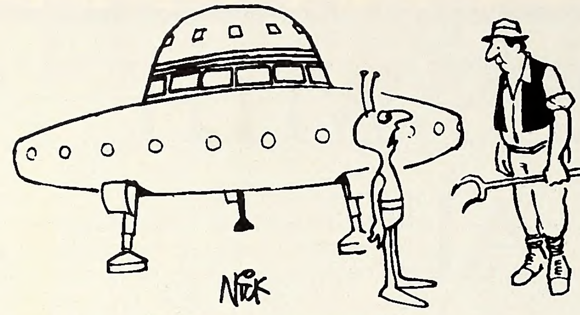
Saturn's 32K RAM boards retail for $239 apiece, and the *VC Extend* software goes for an additional $100.

On the subject of memory expansion: it's worth knowing that even if your Apple is full of RAM boards, there's a maximum worksheet dimension imposed by the *VisiCalc* program itself, not by the extent of memory available. To discover this for yourself, try the following:

Clear a worksheet and enter something in A1. Now go to row 200 and see how far to the right you can enter a value. Somewhere around column AS (if you're using *VisiCalc* 3.3) you should get a beep, indicating that you've gone beyond the maximum worksheet perimeter allowed by the program. You won't be able to enter anything to the right of or below this point, although there's plenty of memory left.

If you clear the worksheet and try the same experiment on some row other than 200, you'll arrive at a different lower right-hand limit, but in any case the maximum worksheet area will be on the order of 9000 coordinates for the sixteen-sector program and about 12,000 for the DOS 3.2 version.

Once you've staked out a worksheet of maximum area, you'll find the allowable perimeter is fixed, even if you should sub-



"They sent our baggage to Jupiter"

Nick/The Bulletin/Sydney/Reprinted from World Press Review/November 1980

sequently blank the entry in the lower right-hand corner. That is, if A1 and AS200 define the corners of a maximum-area rectangle, then AR201 is out of bounds, even if you decide to erase what you had on AS200.

This does not mean that you could ruin a live worksheet by accidentally putting something into some far-removed coordinate. What you would need to do in that case is erase the errant entry, save your work to disk, clear the worksheet, and reload. At that point the program will have forgotten your mistake.

The maximum worksheet area is just one of those things the *VisiCalc* manual fails to mention. Of course, under ordinary circumstances—without such things as RAM extenders—it's not likely that you would run out of space before running out of memory, although it could happen if you were deliberately leaving a lot of open territory between elements of your worksheet.

Another topic completely overlooked by the *VisiCalc* manual is the storage of *VisiCalc* commands in /PD files for deferred execution. Aurora Systems (Madison, WI) has documented this aspect of *VisiCalc* in a set of utilities called *Versacalc* (see the review in this column, September 1981), and has used the /PD exec-file technique to create an item called the *Performance Manager*.

Performance Manager is a set of interacting *VisiCalc* models tied together by a central menu. The menu, too, is a *VisiCalc* model. To run the program, you first boot your sixteen-sector *VisiCalc*, then do an /SL PERFORMANCE, whereupon the menu appears on your screen. Menu selections are also made with /SL commands; the first files you choose from are named, simply enough, A, B, C, D, E, and F. These various models—and the submodels they call—are designed to overlay each other in the right places and pass information between them. At any time, you can return to the menu with an /SL MENU.

In other words, the *Performance Manager* runs within *VisiCalc* but is designed to work as much as possible like an ordinary financial data-base program.

Aurora provides a two-sided disk, one side for a 48K Apple and the other for an Apple with 64K or more. They differ only in the number of budget categories you can track. The 48K version will do ten; with 64K, you get twenty-three.

The program lets you specify a budget amount or goal, for each of twelve months, for each of your ten or twenty-three categories. The categories can be anything trackable in numbers on a monthly basis—expense or income categories, employee or department production levels, or whatever. You also have the option of entering historical data for the previous twelve months, if such data is available.

At the end of each month, you enter data for each category, and the *Performance Manager* compares your actual figures to three other quantities—the projected figure for the month, the actual figure for the same month of the previous year, and a moving average derived from the previous twelve months. Each comparison is reported as an absolute variance and a percent variance. Year-to-date totals are shown as well, and compared to the year-to-date projection and the year-to-date value as of the same month of the previous year.

All this information is tabulated to print on a 132-column printer; eighty columns will do fine, though, if you don't mind printing in segments.

Like *Versacalc*, the *Performance Manager* makes use of split-screen *VisiCalc* models; one side of the screen is for data entry and the other for instructions. Because the models are so thoroughly prompted, it's pretty hard to go wrong—unless, of course, you want to get creative and ignore the instructions.

As with any kind of *VisiCalc* template software, a certain amount of creativity and customization is possible. However, because the *Performance Manager* uses exec-file programming techniques in the manner of *Versacalc*, users not thoroughly familiar with *Versacalc* will probably have a hard time modifying anything fundamental like the dimensions of the screen or locations of particular elements.

The *Performance Manager* retails for $75.

Turn your Apple into the world's most versatile personal computer.

The SoftCard™ Solution. SoftCard turns your Apple into two computers. A Z-80 and a 6502. By adding a Z-80 microprocessor and CP/M to your Apple, SoftCard turns your Apple into a CP/M based machine. That means you can access the single largest body of microcomputer software in existence. Two computers in one. And, the advantages of both.

Plug and go. The SoftCard system starts with a Z-80 based circuit card. Just plug it into any slot (except 0) of your Apple. No modifications required. SoftCard supports most of your Apple peripherals, and, in 6502-mode, your Apple is still your Apple.

CP/M for your Apple. You get CP/M on disk with the SoftCard package. It's a powerful and simple-to-use operating system. It supports more software than any other microcomputer operating system. And that's the key to the versatility of the SoftCard/Apple.

BASIC included. A powerful tool, BASIC-80 is included in the SoftCard package. Running under CP/M, ANSI Standard BASIC-80 is the most powerful microcomputer BASIC available. It includes extensive disk I/O statements, error trapping, integer variables, 16-digit precision, extensive EDIT commands and string functions, high and low-res Apple graphics, PRINT USING, CHAIN and COMMON, plus many additional commands. And, it's a BASIC you can compile with Microsoft's BASIC Compiler.

More languages. With SoftCard and CP/M, you can add Microsoft's ANSI Standard COBOL, and FORTRAN, or Basic Compiler and Assembly Language Development System. All, more powerful tools for your Apple.

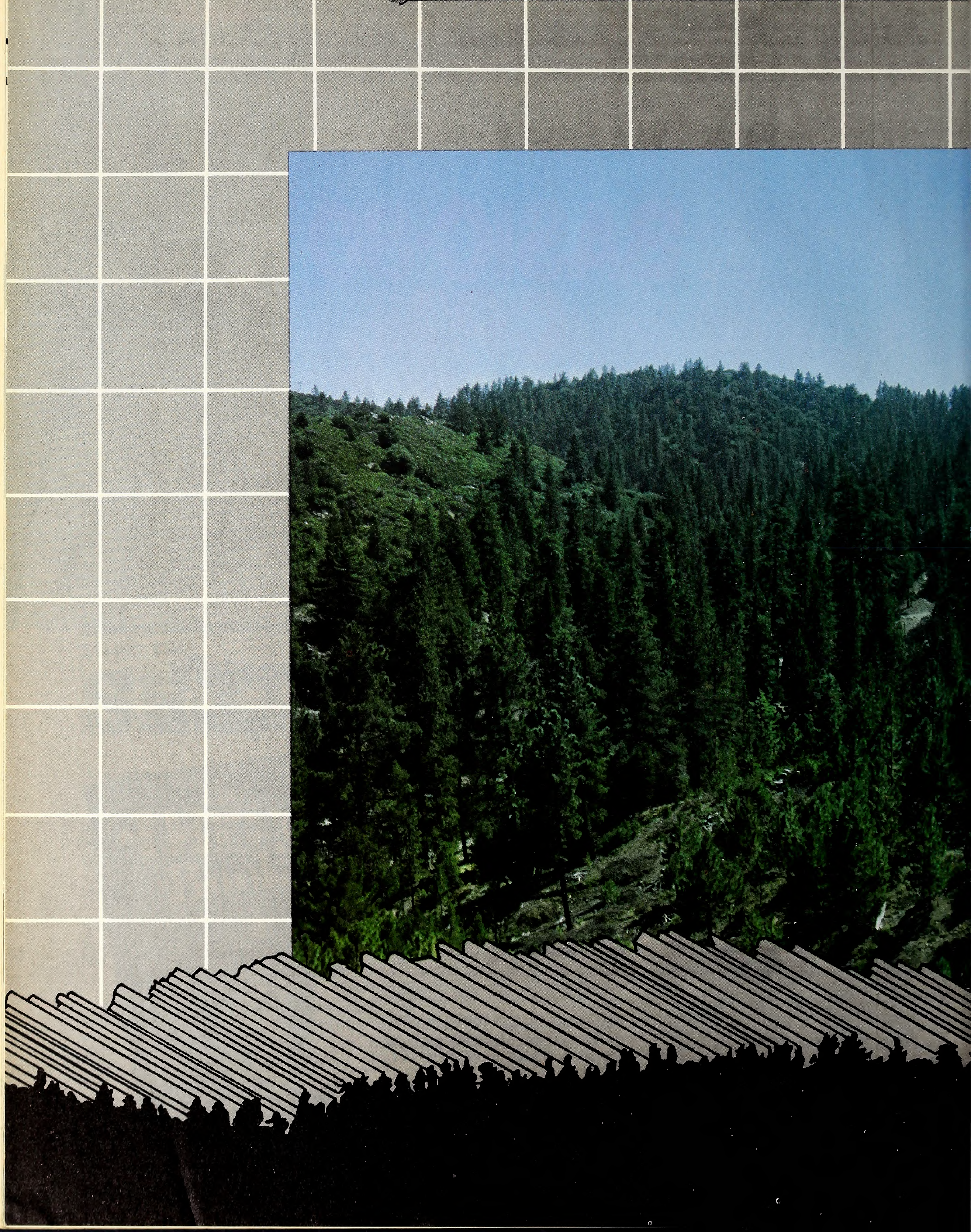
Seeing is believing. See the SoftCard in operation at your Microsoft or Apple dealer. We think you'll agree that the SoftCard turns your Apple into the world's most versatile personal computer.

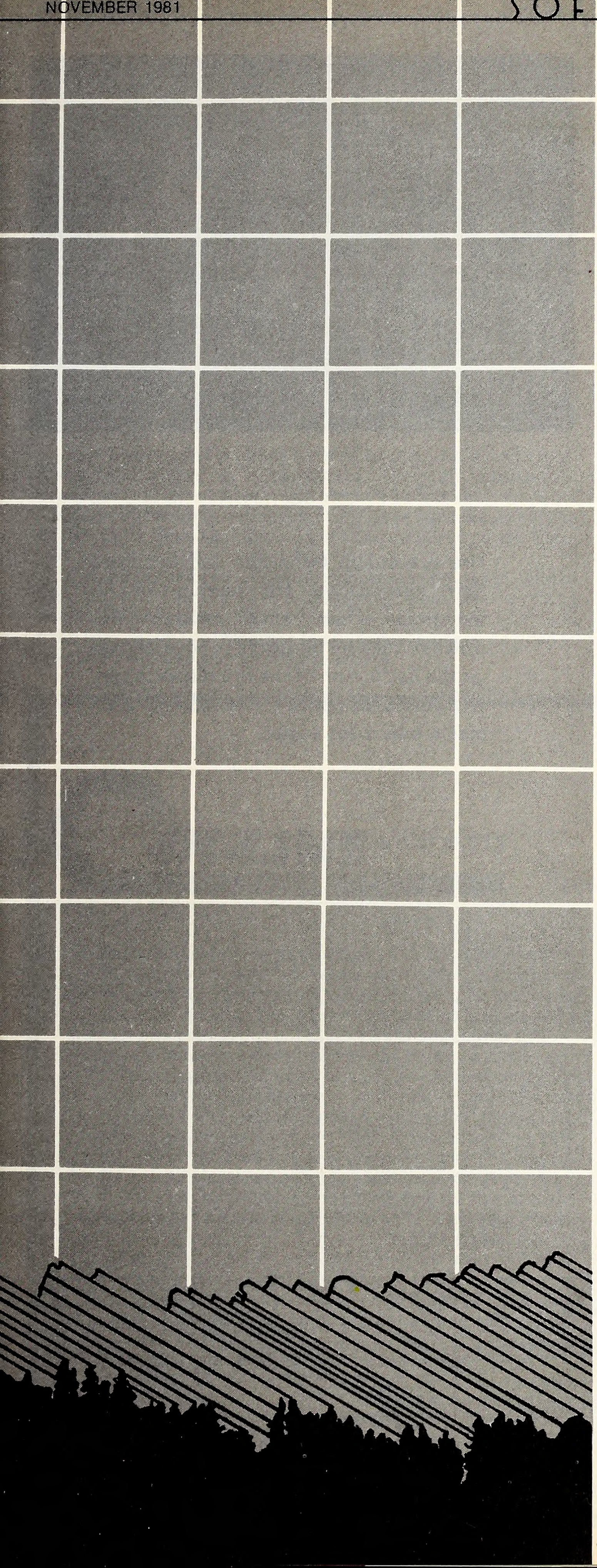
Complete information? It's at your dealer's now. Or, we'll send it to you and include a dealer list. Write us. Call us.

SoftCard is a trademark of Microsoft. Apple II and Apple II Plus are registered trademarks of Apple Computer. Z-80 is a registered trademark of Zilog, Inc. CP/M is a registered trademark of Digital Research, Inc.

MICROSOFT
CONSUMER PRODUCTS

Microsoft Consumer Products, 400 108th Ave. N.E., Bellevue, WA 98004. (206) 454-1315





Apples Among the Acorns

BY MELISSA MILICH

In the beginning was the forest. Even in the earliest autumn, the trees thrust their painted leaves skyward, reaching for the last rays of warm sun before the winter freeze. Just as the oaks and maples shed their cloaks, the firs began to thicken with warming needles. Time couldn't disturb the forest.

The oldest living thing on earth is a tree. A single bristle cone pine has celebrated nature's autumn ritual for more than four thousand years. Before the Pilgrims landed, before the Roman Empire fell, before the Bible was written, a seedling in southern California pushed through a layer of dolomite, and the full-grown tree that it became has survived forty-five hundred autumns.

Now it welcomes another. Woods thick with green break suddenly into brilliant yellows, scarlets, and ambers; hillsides are dotted proudly with flaming Indian paintbrush or heather, spiny yucca or mossy clover, and rusty orange buckwheat or pale, sweet winter alfalfa. The sunshine softens, and the wind nips gently.

Autumn in the southern California forests might be the most beautiful time of the year, but the glory of the season is laced with danger. Fall brings the peak of the fire season.

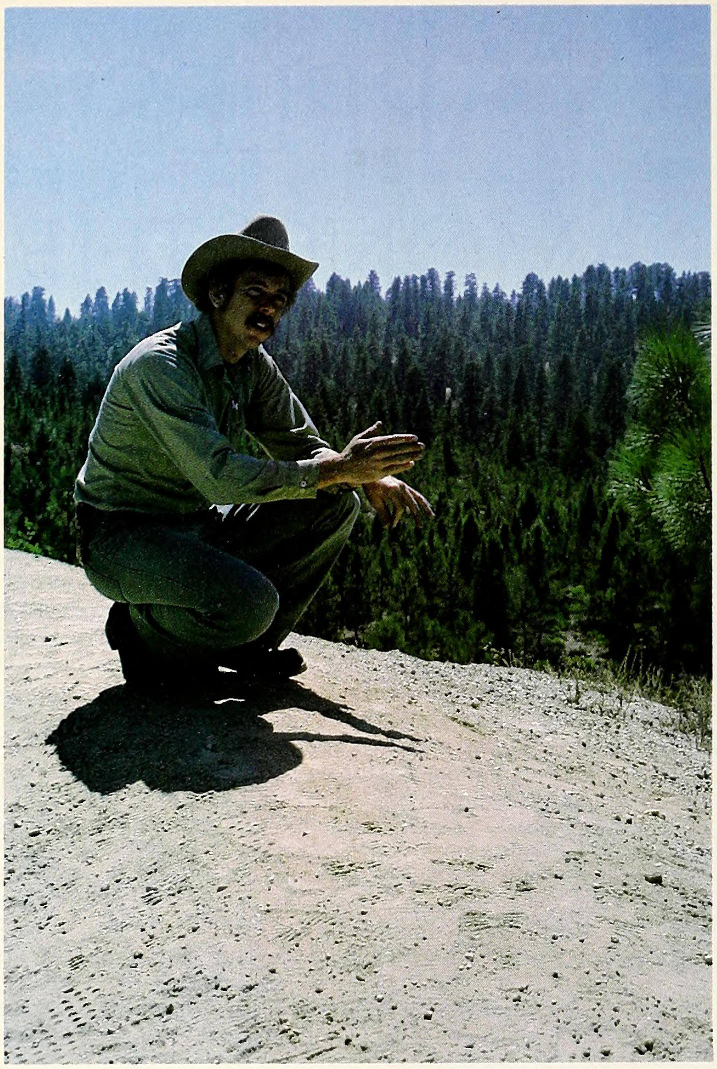
Out from the stillness of the forest bumps a government truck driven by a ranger at work. Jeff Bradford has been with the United States Forest Service eight years and, at twenty-seven, is already an old-timer. Bradford momentarily lifts his foot off the gas pedal and points off in the distance. "A tree's just so doggone pretty, and it's so scary to see it burst into flames. It goes just like a stick of dynamite."

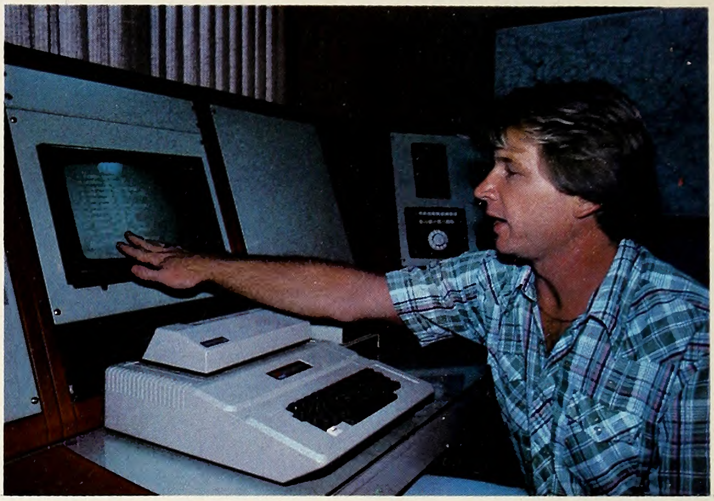
A Lot of Responsibility. Technically, Bradford is not a forest ranger. His working title is fire prevention technician but, because he's wearing a green uniform and driving a green truck, the public thinks he's a ranger. His main job, however, is preventing fires. Out of the 690,000-acre Angeles National Forest, approximately twenty-five thousand acres are his responsibility.

Bradford works for a branch of the forest service that blends the old and new. The timeless beauty of the woodlands is preserved as carefully as possible through fire prevention education for the public and conservation programs designed to make the forest last for at least another million years. The forest service has been doing all that and more since the first ranger station was constructed in the Shoshone National Park in Cody, Wyoming.

One feature of the service is only five months old, but it's already making Jeff Bradford's job a lot easier. Yet Bradford has only seen it once.

Dispatching Apples. Far away from the clean air of the Angeles forest is a glass-walled computer room that looks out on the smog and traffic of greater Los Angeles. This office in downtown Pasadena is the main dispatch center of the An-





Fire prevention technician Jeff Bradford (left) appreciates the serenity of his mountain office, while assistant dispatcher Michael Bergdahl (above) thrives on the hectic rush of the fire dispatch center. Apple computers are making both men's jobs a lot easier.

Debugging the Forest

On Nantucket Island, Massachusetts, the wing tip moths are deforming and killing the leaves of the red maple. In the Kisatchie National Forest in Louisiana, the pine beetles are living high on tree bark. Meanwhile, armies of fire ants in the south are marching north, ready to plunge their stingers into any animal or human that gets in their way.

Apple computers are helping to keep the forests green in more ways than one. The United States Forest Service's southeastern region keeps seven Apple computers busy, devoting one entirely to pesticide management. It's an important task; when that pine beetle gets started, it can kill huge lots of trees.

The computerized pesticide program works like this: A forest crew calls the regional office and is hooked up to a data base, which calculates the type of pesticide to be used and dilution formula. The number of acres to be sprayed and nozzle type used determine the formula.

But debugging the forest is the job of only one Apple. There are six more micros that staff forester John Allen says are used in a variety of ways, including sending out fire fighting information, much as the southern California region does. The Atlanta Apples are hooked to a Hayes Micromodem, and most of the national forest ranger stations have at least a dumb terminal. Computer staff can now send out information such as weather reports and fire conditions immediately to those hard-to-reach rangers in the Ozark Mountains or in Louisiana's Daniel Boone Forest.

Allen says the forest service chose Apples because they didn't require much computer programming skill to operate. "In five to ten minutes a new worker in the Atlanta office can be trained to use the message system," he says.

"We keep finding more and more applications for Apples all the time. This year I'm putting the entire aviation budget on it."

One last word about the Atlanta system. When a worker signs off a program, the Apple says, "Good-bye, y'all."

It is, after all, a southern computer.

geles National Forest. Although the center is far from the forest itself, when a fire breaks, life in the dispatch room becomes just as hectic as work in the field.

Michael Bergdahl knows. He spent eight years fighting fires before he moved to his present position as assistant dispatcher. Instead of a shovel and fire hose, he now uses an Apple computer.

Apples are in use by dispatchers in three of the seven forests in the southern California forest service region: San Bernardino, Cleveland, and Angeles. Basically, as soon as the dispatcher is alerted to a fire, he punches in the location on the computer. The resulting display tells the dispatcher the nearest fire fighting crew to send.

Three Minutes Can Save a Dozen Trees. Before the computer system was installed, the initial dispatch process took approximately three minutes; with the Apples, it takes about five seconds. Three minutes might not seem like a long time, but fires can get out of hand fast when there's nobody around to put them out.

"Keep in mind that this is a nonmunicipal fire department. We're fighting fires out in the boonies," Bergdahl emphasizes. Consequently, forest fires might require helicopters, air tankers, and bulldozers to extinguish the flames; and the Apple has stored on its data base the information required to get them moving.

The computer-based dispatch system was the brain child of Bergdahl and Daryl Paige, a dispatcher from the Cleveland National Forest. Both men worked in the field together, and when they made the transition to an indoor job, they decided to use their field experience to help others.

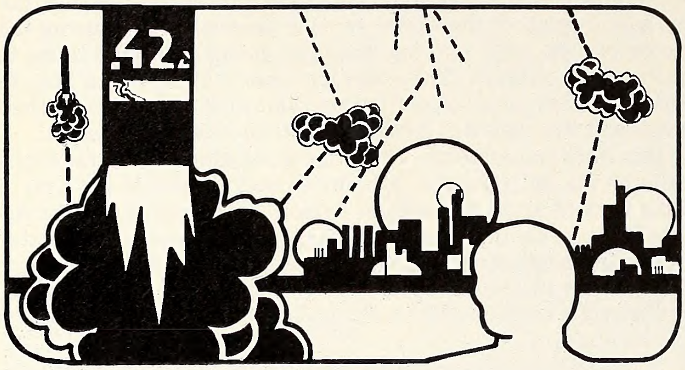
Bergdahl and Paige met with five other dispatchers in the southern region of the forest service several times during the winter of 1980-1981, and together the group compiled ideas to improve the system. They handed over their ideas to Martin Wefald, a computer expert within the forest service, who then developed the dispatch program put into use last June 1.

Bradford, meanwhile, twists his truck along the narrow dirt trails on his daily patrol. It bounces and shakes every time it hits a rock or sinks into a hole—which, in this section of the Arroyo Seco, is about every six inches. "This is one of the better trails," says Bradford. He doesn't have to worry; he has a cup of his wife's mountain coffee brewed with cinnamon in hand, which protects even the least hardy from overwhelming motion sickness.

A Season for Destruction. The fire potential is fairly extreme this autumn because of the lack of rain during the year. June through December is the traditional fire season in the southern California forests, but the coming of the hot, dry desert winds in late September, October, and November makes fall the most dangerous season. Last year, the three largest fires developed in November; historically, if one gets started, a rash will follow.

Bradford's not expecting anything different this year, but Michael Bergdahl, back in Los Angeles, has his own ideas. After all, his office has been completely revamped by the addition of the two new Apples.

The old system was clumsy at best. When a first-alarm fire (not so serious) was called in, the dispatchers would leave the radio area and run to the wall map in another area of the room. Colored pins on the map designated units available and where



they were located. Then they would run back to the radio area and flip through a fat book filled with page after page of topographical information about the area of the fire.

If the first-alarm fire turned into a second-alarm fire (much more serious), then the dispatcher had to run back to the map. Helicopters and air tankers would be called in, and it would be the dispatcher's duty to figure out the air miles and flight directions for the pilots.

If the process sounds confusing or hectic, remember, all that activity was for just one fire. Add three more brush fires in different areas of the forest, plus a lost child, and throw in an injured hiker. Sounds like television, but sometimes it happens in real life.

Apple's Instant Recall Handles Logistics. Now all this information is conveniently stored on the computer. So if a second-alarm fire, three flaming brush areas, one lost child, and an injured hiker all happen at once, Bergdahl and his dispatch crew can handle it. Hitting return on the Apple and then S for send gives them the next available crews to go into action. If aircraft is needed to fight the fire, the dispatcher simply punches in the craft's location, and the computer displays the distance and directions to the fire. A fire at Bear Canyon, for instance, would require a helicopter to fly twenty-six miles and seventy-two degrees north.

Bergdahl rattles off on his fingers twenty-five different engines, thirty-five patrols, two helicopters, two air tankers, three water tenders, one bulldozer, and thirty other miscellaneous fire fighting resources of which the dispatch department keeps track.

"That's a lot of resources. It's a big forest, and there's been nobody yet who's been able to memorize all the areas."

In addition to the computer statistics, Bergdahl says the dispatchers still use buttons and pins on the map to represent resources. Magnets correspond to people. "But that was all we used to have for keeping track," says Bergdahl. "We'd get a line-up in the morning, and we'd place a magnet wherever a crew would be for the day.

"But then maybe engine fifty-five would call up and say he was unavailable on his unit, so we would store that in our head, because we couldn't really designate it on the map. If we had twenty calls like that of these engines moving around, we couldn't remember where anybody was."

Bergdahl says it takes him a few minutes in the morning to update the program for the day. Now if the dispatchers want to know where engine fifty-five is located, they ask the program. Bergdahl demonstrates it by going to the computer. He hits a key, and five seconds later it shows that engine fifty-five in district five is at the station.

"There's always been a need for quicker and better service," Bergdahl says. "Five seconds with the computer compared to three minutes running around—and you'd better believe that we use that extra time we save."

Matching Specific Fire to Methods that Worked Historically. How do the dispatchers use that extra two minutes and fifty-five seconds? With more running around, of course. However, now they run into an adjoining room to another computer terminal connected to a main information center in Atlanta, Georgia, which contains fire fighting research material on disk.

Bergdahl and his crew now have time to obtain useful fire fighting statistics that they can send to the crew in the forest. By punching in the location of the fire and the weather conditions for that day, the computer will print out projected rate of fire spread, flame length, and increasing heat constants—all information that the dispatcher can radio to the fire staff on the way to the fire, so they can decide whether they need more engines.

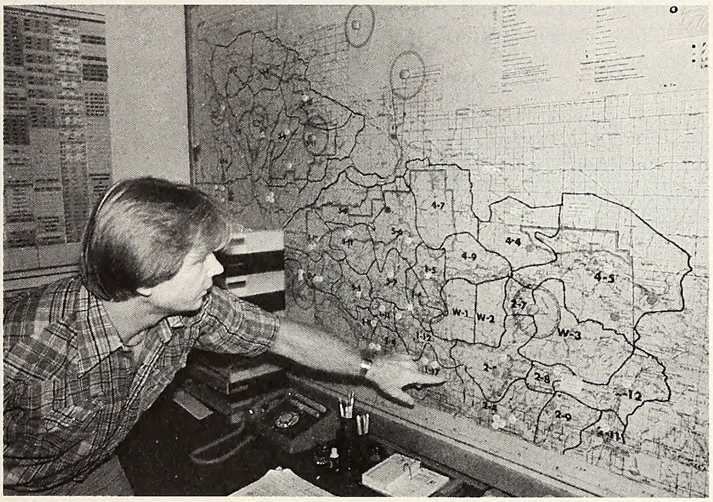
"Believe me, there's nothing like these computers," Bergdahl says. "We can get a lot more information to the troops now because we have more time. Last year, we couldn't have gotten this out. No way."

Out in the field, riding along on his roller coaster-like route, Bradford admits that he doesn't think about the computers

very often. "We're aware of them, but way out here we can't easily see the difference they make.

"But if I need information, I know they can get it to me fast. Whether it's fast because of the computer or just because they're efficient and organized, it's fine with me.

Apple Enhances Interdepartmental Understanding. In some ways, Bradford is not a stranger to computers. His father, James, started working with computers in 1955 as a programmer and consultant for a private company. But Jeff, at least for the time being, seems fairly content out in the field. He smiles about the time he once visited the computer room in Pasadena and knew right off that he was out of his milieu.



Softalk photo

This is the wall map dispatchers used to rush to when figuring out locations of their fire crews. Dispatchers still use the map, but much of its information is now stored on the Apple.

"Mike [Bergdahl] showed me how to push the button and get the location on the screen. It was pretty amazing, but I was glad to get back to the woods."

Bergdahl would like to see more of the field workers visit the dispatch center. Their meeting usually helps field workers and dispatchers understand each other's frustrations. "It really helps our rapport," he says.

"A barrier tends to develop on both sides. But when they come in from the forest and see how busy and hectic it can get in here, they no longer wonder why we're not on the radio talking to them every minute.

"For example, inevitably you'll get caught in here by yourself," says Bergdahl, "or with only one other dispatcher, and you're already short-handed. This program, however, helps one person, even if he's not too familiar with this forest, to work Angeles dispatch."

With or without computers, Bergdahl says few field personnel wish to return to dispatch. He thrives on "the stress and the activity in here, but some people can't handle it," Bergdahl shrugs.

Margin for Error. The computer program developed by Martin Wefald of the forest service's San Francisco regional office is a genius program as far as Bergdahl is concerned. Before the computers, there was a large margin for human error inherent in initial dispatches. Now the program allows its own margin for error, and Bergdahl tests it every day with his typing.

During major fires—such as the 130,000 acre Marble Cone fire in the Los Padres Forest in 1979—crews and dispatchers from out of state are sent in to help the southern California division. Wefald designed the program to be easy for outsiders to run; it takes only a few minutes to learn. And since out-of-state dispatchers would not be familiar with the 650 sections of the Angeles forest, he installed a feature into the program that allows for spelling errors.

"It's phonetically oriented," explains Bergdahl. "It'll correct my spelling mistakes." To demonstrate, he typed in "Bukke Bay" for the location Bouquet Bay, deliberately misspelling it. The computer wasn't fooled. It listed everything in its catalog that phonetically sounded like "Bukke Bay."

"We may employ three hundred seasonals a year, which means they're working only during the fire season. They may come from Ketchikan, Alaska—who knows? But if they've radioed in a name they don't know how to spell, the program doesn't suffer from it."

Bergdahl pointed proudly at his misspelling of Bouquet Bay. Typically, he knows the correct spelling for the names of the forest, but sometimes his two-fingered typing lets him down, and he makes more errors. Again, the computer appears to understand the need for urgency in the situation and makes the corrections.

"See, I even spelled *ranger* wrong, and it caught that. I also did that intentionally."

Bergdahl stopped abruptly and peered closely at the computer screen. "What do you know? I spelled *station* wrong, too. Oh, well, the computer fixed that also."

The Serenity of Readiness. The forest is in good hands. Martin Wefald is in San Francisco working on more programs the government might adopt; Bergdahl is putting them into practice in the dispatch room; the computer is taking care of spelling errors; and Bradford is driving around in his truck watching out for fires and admiring his outdoor office.

"The forest is so pretty this time of the year," says Bradford. He drives past the stomping grounds of Grover Bear (no relation to Smokey), who has been dumping out the contents of and bashing in a government garbage can.

The truck rambles along. "Look, there's mountain mahogany. That over there is called skunk cabbage. And that big red plant is poison oak."

Bradford loves his forest. And with the help of wildlife conservation, fire prevention programs, and microcomputers, it looks like it'll stay around for a long, long time. ∎

Exec Brøderbund

From page 59

seemed that the company's place as a major publisher of entertainment software was secure.

That magnitude of growth caused the brothers to look around for someone to share in the business management. Sister Cathy came to mind, and they broached the subject to her.

Cathy had found that concomitant with the glamor of being a New York buyer came the tons of paperwork that preceded and followed every decision. She also had come to the conclusion that Manhattan was not where she wanted to spend the rest of her life.

So when her brothers importuned her, she responded with the warning that if they weren't serious, they'd better drop the subject because she just might take them up on it.

When they persisted in other conversations, Cathy, who didn't know a RAM chip from a disk drive, decided she'd better investigate what her brothers were up to. She dropped in to the Computer Factory in New York and came away impressed with the energy of the microcomputer industry. Whereupon she joined up, coming on board just in time for this year's Computer Faire.

A Sister's Place in Broderbund. She's applied her learned skills in such diverse areas as advertising, fair and exposition participation, and accounting. These were potential problem areas for a former attorney and a former basketball coach, but her savoir-faire in coping with these arcane areas has enabled the company's growth to continue apace.

Because of Broderbund's entertainment orientation, it was with some bemusement that the industry observed Gary Carlston's announcement that the company would be releasing a payroll package. Even he admits that there was no compelling reason, other than the quality of Hal Faulkner's package, for Broderbund to get involved in the business area.

But Faulkner had a well tested payroll program; and even though it was not a part of an integrated accounting package, its virtues commanded Gary's attention. *Payroll* made its debut at the Computer Faire and has since proved to Gary that Broderbund can find a niche in the business market. Meanwhile, Faulkner is putting the finishing touches on a general ledger package to serve as an accompaniment to *Payroll*.

The major emphasis remains on entertainment software, however, with several new packages in development simultaneously now that the company has added Chris Jochumson to the staff. *Space Quarks* is his first published effort for the company.

And Now for Some Programmers. . . . Jochumson is a genuine, 24-carat assembly language programmer who now works in conjunction with Doug Carlston. Doug develops the concepts for the software and Jochumson implements them. Jochumson is presently juggling three program concepts that may see the marketplace before Christmas.

Broderbund has also broadened its publishing base by acquiring product from Marc Goodman and Olaf Lubeck, both authors of successful Apple software for other publishers.

This new spate of activity caused the Carlstons to reevaluate their Oregon location. Deciding they were somewhat removed from the center of the action by being so far from Silicon Gulch, and recognizing that their Star Craft connection was more conveniently served from a major city, they began casting about for new headquarters in the San Francisco Bay area.

They're now ensconced in a hilltop home in San Rafael that overlooks one of the last buildings ever designed by Frank Lloyd Wright—the Marin County Civic Center.

From that site, the Broderbunders continue to disprove Leo Durocher's baseball maxim that nice guys finish last. ■

TIME AFTER TIME

BY RICHARD KAAPKE

In the September Softalk, we discussed three real-time clock/calendar accessory cards for the Apple. Now it's time to take a look at two other real-time clock cards: the Model 7424 Calendar/Clock Module from California Computer Systems (Sunnyvale, CA), and the Time Machine II from Creative Software Development (West Valley City, UT).

In case you don't recall, clock accessory cards enable the computer to know the time and date without asking for it—very handy in many business and scientific situations. Many real-time clock cards can generate interrupts—which means they have the ability to get your computer's attention as easily as an alarm; they allow the computer to appear to do two things at once.

CCS's Updated Calendar/Clock. The CCS Model 7424 Calendar/Clock Module is one of the older clock accessory cards for the Apple, yet recently revised firmware (special machine control instructions programmed into the card) gives the new model 7424 more abilities than its predecessor.

Unlike some other clock cards in its immediate price range, the model 7424 does not include the necessary hearing-aid type batteries, nor is there any software included on disk. The software that comes with the card is in printed form for you to keyboard.

After purchasing the batteries (and perhaps a disk), you'll be faced with reading a manual that has a heavy technical style, full of jargon only experienced electronic engineers and seasoned computer owners will find easy to follow. Most of the confusion users are apt to feel reading the documentation

stems from the lack of illustrations and explanations telling how to find and install the option jumpers on the card. People purchasing the model 7424 who cannot identify jumpers or install and remove them are advised to have their dealers set up their cards and show them how the jumper installation process is done.

Once the jumpers are in, the model 7424 is ready to be used. A program in the owner's manual allows you to set the time on the clock. The revised firmware now included with the model 7424 will keep the hour, minute, second, day of the week, date, month, and year—all available at your Apple's request.

The model 7424 can return its information to the computer in the following ways: clock input, time string, and screen display. If you don't understand these terms, read on.

Learning to Tell Time. The clock input routine is the usual method of reading time from a clock accessory card, regardless of manufacturer. You'll find this method has been incorporated into most programs that read time from clock cards (business software, filing programs, and so on). The trend was set by the first clock for the Apple, appropriately called the Apple Clock, from Mountain Computer.

Basically speaking, the instructions

```
IN# 4
INPUT x$
IN# 0
```

are supposed to return the time stored in the string variable x$ according to the following format:

MM/DD hh;mm;ss.xxx

where the placeholders MM stand for the month (a number from 01 to 12), DD for day (01 to 31), hh for hours (00 to 23), mm for minutes (00 to 59), ss for seconds (00 to 59), and xxx for milliseconds (one-thousandth of a second, 000 to 999).

Our example assumes that the clock card is installed in slot 4, hence the IN#4 instruction. This example will work with the CCS model 7424; however, the last field (milliseconds) will always read 000, because the model 7424 doesn't measure time to the split second.

Stringing Along. The time string routine is a special way of stuffing a value into a string without having to program the way we did in the example. The Basic instructions in the example must be run every time the current time is needed. With the time string routine, a simple set of instructions will set up the computer so that the string variable TI$ always has the correct time. This bit of magic is done using the model 7424's interrupt capability to reset the contents of the variable TI$ every second. This is a simple example of the two-things-at-once feature that interrupts have: a program can be running using time information from variable TI$ while another program keeps TI$ up to date and seems to be running at the same time.

The screen display routine is the 7424's last timekeeping function. Using this capability displays time on the screen of the Apple until the routine is disabled. There is no simple way for a program to read the time off the screen, so the display is assumed to be for aesthetics only.

The CCS model 7424 returns year and day-of-the-week information only in the clock input routine; one Basic instruction can switch this information on or off. The three routines of the CCS model 7424 are selected by hardware (adding or removing jumpers), and it's wise to turn off the Apple to change modes.

There are four interrupt rates: 1,024 times a second, once a second, once a minute, and once an hour. It's a good buy for a simple timekeeping accessory; $125 is the lowest price for a clock accessory card.

H. G. Wells Rides Again. Time Machine II from Creative Software Development is a flexible clock accessory card for the Apple II. The operating manual is well written, quickly getting you through clock installation and then issuing simple instructions from the keyboard to read and set the clock.

Where the CCS clock requires the movement of jumpers to change modes of operation as well as interrupt rates, the Time Machine II can change modes and interrupt rates with two Basic instructions. It keeps the hour, minute, second, day of the week, date, month, and year; interrupts at rates of 1,024 a second, once a second, once a minute, and once an hour can be requested with two Basic instructions.

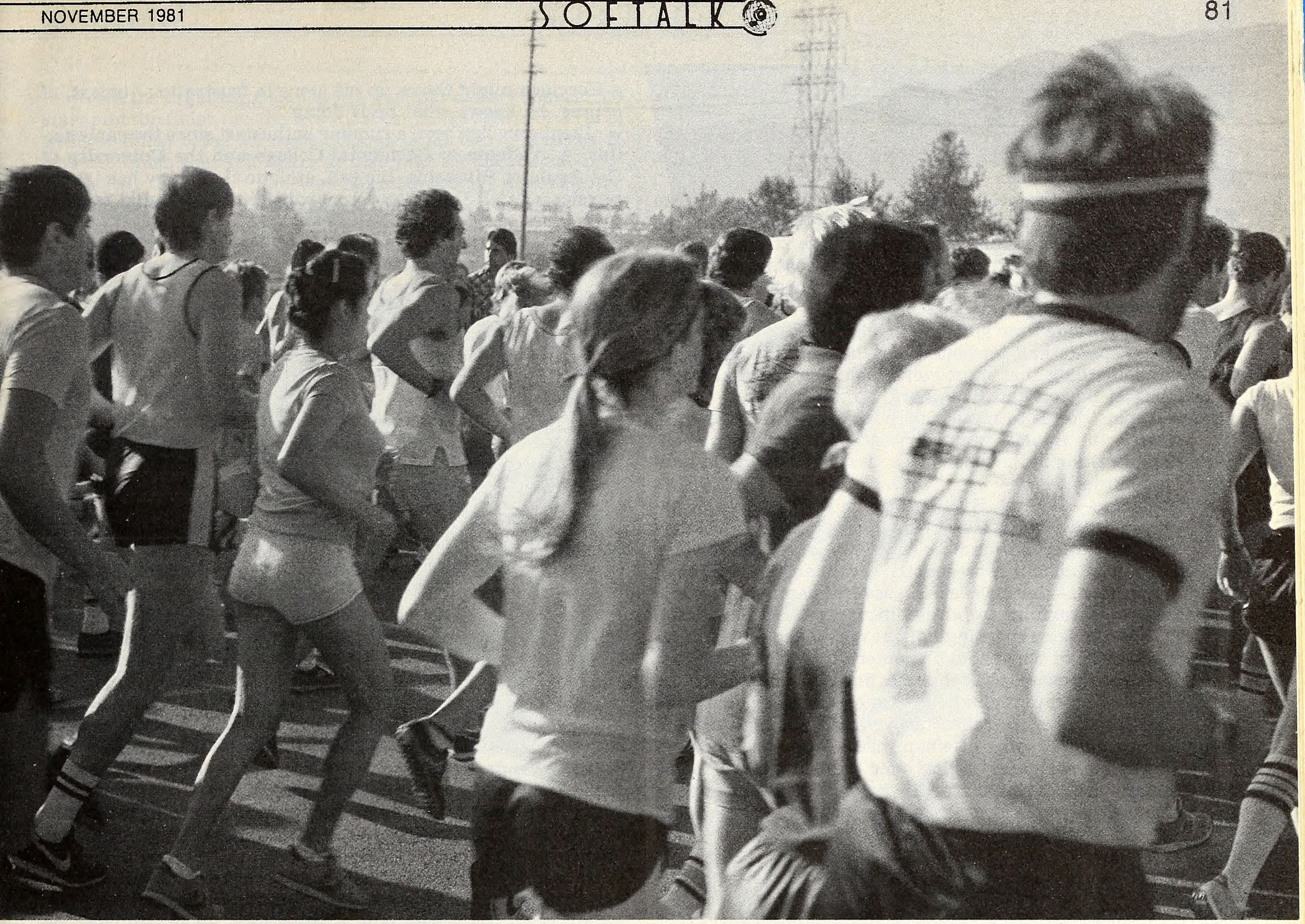
The Time Machine II can perform like the CCS 7424, with the advantage of not requiring rewiring to change modes. It comes equipped with battery and software disk and has an easier mode of operation than the CCS board. Another difference is the return of the time: the format Time Machine II sends is

MM/DD hh:mm:ss.WYY

where colons have been substituted for semicolons and the placeholders WYY stand for the day of the week and the year.

Time Machine II has six date display modes (one to fit almost any need) and seven time display modes. Clock input routine, time string routine, and screen display routine modes are built-in, with little or no difference from the way CCS 7424 behaves in the same modes. The disk supplied with Time Machine II includes demonstration programs and a clock setting program, plus a datestamp function you can incorporate in your DOS 3.3 disks to mark your files with the date and time the files were last written to or created. For $15, you get a spooler disk that allows you to print files on your printer while running another program. Another $15 buys a disk with the programs necessary for using Time Machine II with Apple Pascal version 1.1.

At $135, Time Machine II appears to be as good a value as any unless you require the special interface capability offered by Mountain Computer or the Thunderclock.



Five Thousand Runners
RACE AGAINST THE APPLE
BY DAVID HUNTER

"You've got one hour; everything has to work right then," explains Dennis Ikenberry, founder of Race Central. Based in Rialto, California, Race Central is a small successful company that provides computerized results for professional and amateur distance runs.

On September 13, 1981, outside the Los Angeles Zoo, Ikenberry, his wife Judy, three Apple computers, assorted hardware, and twenty employees faced their toughest challenge. More than five thousand runners would start the NBC Peacock ten-kilometer run, and most would finish within twenty minutes of each other. If computers or humans made mistakes during those crucial twenty minutes, it could be disastrous.

Open Twenty-Four Hours. The widespread popularity of running is a relatively new development. Ikenberry calls it a "completely startling phenomenon. A long time ago, in the early sixties, one hundred people was a lot for a race."

In this age of emphasis on fitness, many people find running the easiest way to keep in shape. There are no lines to wait in, as there are for tennis or racquetball courts; there's no requirement for a partner or opponent; and the hours of availability are unlimited.

On the negative side, *where* to run can pose a problem. Tracks are boring places to run; along roads, runners are subject to intrusive comments from passersby and motorists; and public parks are inconvenient if they're not close to home.

Although he foresees a steady turnover in the running population, Ikenberry believes the popularity of running and racing will continue as long as physical fitness remains a priority value. Even if the trend should change, a core of devoted runners will go on agitating the gravel.

For the first five hundred people finishing in an event the size of the Peacock race, running is almost a religion. The first official finisher, entered in the twenty to twenty-four year old males category, was Kurt Pfeffer, a veteran marathon runner.

The majority of runners at the Peacock run were not out to win the race but to finish it as best they could. Since it would be unfair to measure everyone against the best runners, the Peacock run has twenty-nine different runner categories, determined by age and sex.

Whether dedicated professionals or not, most runners like to see their results as soon as possible. In the Boston and New York marathons, the results are mailed to the runners weeks after the event. This is where Race Central beats everybody.

Race Central's job is to end the race in an orderly and efficient manner, to keep track of every runner and his or her time, and to provide time and ranking results for the majority of the participants half an hour after the race. Race-site results are something that most runs, including the mammoth marathons in Boston and New York, do not provide.

The Apples' job is to make this possible. With the help of two printers, the Apples produce results that show each runner's time, position in the whole race, and position within individual categories.

Side by Side—At First. By 8:00 a.m. on Peacock Race Sunday, quite a crowd had assembled. Photographers shouldering television cameras roamed everywhere amid much noise and confusion. Runners representative of nearly every variety of human form—there were even several handicapped participants—were warming up, stretching, preparing for the run. The hazy Los Angeles morning light revealed thousands of individuals brave and bold enough to enter athletic competition.

Suddenly they were off, a human flood of bobbing heads, swinging arms, and pumping legs. Running side by side were professionals, who enter races the year round, and less accomplished runners, some of whom had never raced before.

While the runners are jogging over the course, Ikenberry and company are gearing up for the finish. When that flood of people comes roaring in at the rate of ten or fifteen per second, a Hercules might throw up his arms in frustration—unless, of course, he knows what he is doing.

Ikenberry has been a running enthusiast since the early sixties. A graduate of Occidental College and the University of California at Riverside, the tall, athletic Ikenberry has taught physics at Cal State San Bernardino since 1965 and teaches there now part-time.

While serving as head coach of the Southern California Road Runners, a local running club, Ikenberry established the

Running Center, a store catering to the needs of runners. It opened in December 1977. Until this year, the store occupied much of Dennis and Judy Ikenberry's time, perhaps too much. They sold the store last April to devote more time to the race results business.

In late 1978, Ikenberry held a race that drew about a thousand runners. "It took forever to get the results," he says, "and I knew there had to be a better way." A month later, he met veteran programmer Randy Hyde and, through him, the Apple computer. Hyde recommended the Apple to Ikenberry, noting the features needed most: reliability, expandability, and mobility.

"The very first Apple Randy got for us was the one he wrote the *Lisa* assembler on," boasts Ikenberry, who now owns three Apples.

Soon after he began using the Apple, Ikenberry discovered that one of his students, Stuart Boden, was a computer programmer. Using Pascal, Boden wrote programs for the Apple, and Hyde designed hardware for interfacing the Apple with timing devices.

Chuting the Breeze. The procedure for timing a foot race is rather complicated. Each runner has a number and is entered into the computer in an appropriate category. As the runners finish, their times are recorded on a Super-Kim, which stores the times for later matching with the names, which have been stored on the Apple.

After crossing the finish line, the runners enter chutes in the order in which they finished the race. When they reach the end of the chute, each is given a number that matches the time stored on the Super-Kim. It's crucial to make sure that the runners enter the chutes correctly and get their numbers.

It took time to get programming bugs out of the way, and a really good system didn't come about until the fall of 1979. A constant concern was the need for a system that wouldn't crash in the middle of the race. It's not the same as working with a program in the comfort of your office, where equipment failure can be handled (or ignored).

The success of a race hinges on a lot of little things in Ikenberry's line of business. He knows them well.

"It's a real organizational thing. It's also important to know something about running and how runners feel."

From the organizing standpoint and that of the runners, the start and the finish are the two most important parts of a race, with the finish being the more difficult of the two to manage. It was the finish that fell apart at last year's Peacock race.

Fiasco! The finish line was handled poorly, with runners stopping a hundred yards from the finish line and walking into the chutes, sometimes six abreast. Although Race Central didn't handle the finish, the company appeared culpable because it couldn't provide results as promised. In most cases, it was virtually impossible to determine who finished when.

Consequently, there was some doubt that Race Central would be chosen to handle this year's Peacock race. But NBC, influenced by a reassuring newspaper article about Race Central, decided to give Ikenberry a chance to improve on the previous year's fiasco. This time, though, Race Central would handle the finish line as well as the timing and results.

Still, NBC wasn't taking any chances. On their televised special devoted to the Peacock run, aired that afternoon, they interviewed dozens of personalities and covered just about every angle of the race, but they made no mention of Race Central. Ikenberry was not too concerned with the low profile. Just seeing the system work well with so many runners was satisfying enough.

Gene Walsh, vice-president of press relations at NBC, had mostly good things to say about Race Central this time around. "The results were fast and accurate, and it was a very good finish." With more than fifty-two hundred registered runners, the Peacock commanded one of the best runner turnouts in southern California. Although he's very pleased with the good job done by Ikenberry, Walsh is still not completely satisfied, citing the need for more refinements in the system.

From Ikenberry's point of view, the NBC race was special because it showed that the system worked with so many runners. The Race Central system—including three Apples, Super-Kim, hardware, and software—is such a success that Ikenberry and Hyde have visions of selling the system to race directors throughout the country.

Passing the Buck. Race Central is a growing business that wouldn't be where it is if there had never been a company called Apple Computer.

One of the most exciting things about the Apple phenomenon is the impetus it has given to individuals and small groups to enter the business world. Race Central hasn't forgotten its beginnings and enjoys helping others who are just starting out.

On October 11, 1981, Race Central will handle the results for the San Bernardino County Sheriff's "Take a Bite Out of Crime" five-kilometer and ten-kilometer runs. Deputy Sheriff Sonja Barber has found organizing a race a much harder thing than she believed it would be, so Race Central has been guiding her through the unfamiliar territory. "They have been very helpful. If we have a problem, we give them a call, and they usually have the answer."

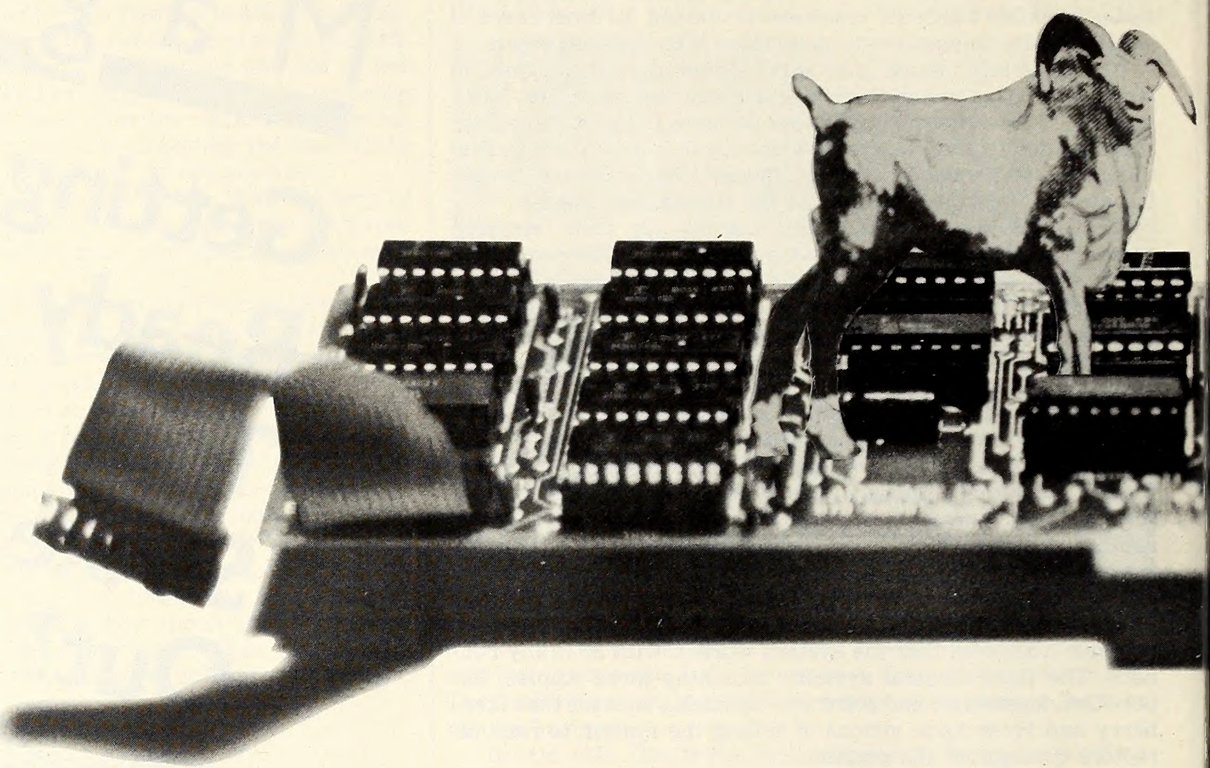
Christopher Meaney, race chairman for the upcoming "To Act: Run in the Sun" in Thousand Oaks, California, is glad he came across Race Central. A fund raiser like many runs, this race has been in planning stages for five months. Nonetheless, all the preplanning in the world won't forestall nagging worries the first time through an experience.

"Most people would say: 'You're new, and if you don't have the money we can't help you.' So we went to Race Central to see what they had, and they gave us a good break."

The Finish Line. Ikenberry was in for a change of pace when he gave up resoling running shoes in his store to try to build a profitable race results business. Throughout the Peacock race, Ikenberry never relaxed, preparing for the hyperactivity at the race's end.

"If anything major goes wrong, it really stacks up on you," he warns. "There isn't any time for recovery when the runners are coming in at ten or fifteen a second."

The NBC Peacock ten-kilometer run was a success for Race Central. Chalk up another victory for a small business that had the foresight to bite the Apple. ∎



Couldn't You Use a Little

PART 1 OF AN OVERVIEW OF RAM CARDS

BY JEFFREY MAZUR

Although it passed without much fanfare, the Apple Language Card recently celebrated its second birthday. Except for the Disk II, probably no other peripheral has had such an impact on the Apple II computer system. Besides its intended function of storing Pascal and other languages, many other uses have been found to take advantage of the extra memory this board provides.

The Language Card was not always so popular. Along with its new capabilities came compromises that have taken until now to resolve fully. Boards sat on the shelf for months because users became frustrated with the many features of the new system. Here's why.

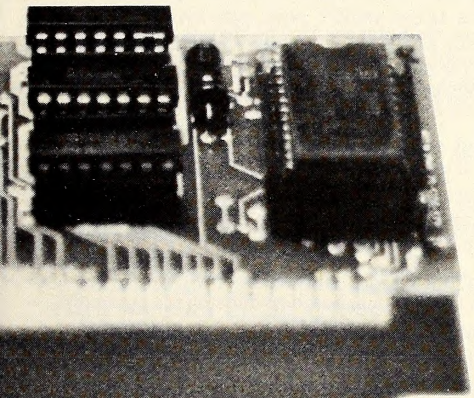
When the Language System first came out, many of its pioneer owners were folks who had been anxiously awaiting the opportunity to run Pascal on their Apples, who'd been following the rumors about Apple UCSD Pascal from the beginning.

At first, it was assumed that the language would be distributed in ROM like the Applesoft ROM card. Then the word circulated that Apple had come up with a better idea—to place 16K of RAM on a board, giving the Apple's 6502 CPU a full 64K of the programmable memory (assuming of course, that you already had the maximum 48K on the motherboard). This extra 16K would occupy the upper address space and was to have special write-protect circuitry so that runaway programs would not clobber it. Also, since the Apple's I/O is memory mapped into the \$C000-\$CFFF address space, 4K of the additional RAM would have to be bank switched. Bank switching allows two blocks of memory to share the same address space with only one accessible to the CPU at a time.

Installation of the Language System was rather involved. One IC had to be removed from the motherboard and a jumper put in its place. Then the board was placed in slot 0. The Applesoft or Integer ROM board, if there had been one in slot 0, was now obsolete. Finally, two other ICs had to be replaced on the disk controller card.

For all this trouble, you got a pretty good Pascal system and an autostart ROM to boot. And, all of a sudden, you had to use a special disk to preboot regular DOS disks. And once in DOS, you had the benefits of the autostart ROM whether you wanted them or not. Want your normal Apple back?

If 95 percent of the time you were using DOS rather than Pascal, the double-booting process probably grew old quickly. Therefore, despite the complicated reconfiguration, some users decided to remove the Language Card, replacing it only



Extra RAM?

when they explicitly desired to use Pascal. This eventually led to cards being left on shelves for quite some time. Soon, however, the rumors murmured, Apple would be announcing DOS 3.3, which was supposed to solve all this.

The New DOS. Like the rumors that had preceded the Language System, expectations about DOS 3.3 had it doing more than it really does. Basically, no changes were made to the DOS itself. The disk format was changed to sixteen sectors, making it compatible with Pascal and increasing its storage capacity. This was fine and dandy, but running DOS 3.2 still required the extra booting of a Basics disk. And, even if you had a perfectly good ROM card filled with Applesoft or Integer, you now had to wait eight more seconds every time you booted while the alternate Basic loaded into the Language Card from disk. With 99 percent of all software still in DOS 3.2, some users decided to wait yet a little longer. Among the earliest Apple veterans, many simply didn't want to give up their old accessible ROM by accepting the autostart.

Well, many fought the changeover to DOS 3.3, but by now only a few die-hards remain 3.3-less. (With such folk in mind, we tried adding a switch to regain use of the motherboard monitor ROM. This turned out to be relatively simple; you'll find the details in part 2 of this article next month.)

Apple sales were still booming, but now only DOS 3.3 was being offered. This gave added impetus to the popularity of the Language System, but many people wished they could get the extra 16K of memory without the Pascal software—and its near $500 total package price.

The demand for the card without Pascal grew even stronger as other uses for the Language Card were discovered. Personal Software's *VisiCalc,* Programma's *Lisa,* and Microsoft's *SoftCard* with CP/M, for example, were able to make use of the added memory if it was available.

About this time, it became evident to several companies that the hardware portion of the Language System—the 16K RAM board—would be a very salable item.

Will the Real Language Card Please Stand Up? The first Language Card look-alike was the Andromeda 16K board. At $200, this was quite a steal—and numerous Apple owners took this literally as a license to steal the entire Pascal System if they had a friend who would give them a copy of the software.

Whatever the buyer's motives, the explosion of RAM boards was under way. In fact, they became such a hot item that Apple finally conceded to the demand by making the RAM card portion of the Language System available separately.

Before discussing the newer 32K and 64K boards, let's examine most of the current 16K boards.

Within the genre of 16K RAM boards, there are entries from companies such as Apple, R.H. Electronics, Microsoft, Andromeda, Computer Stop, Omega Microware, Qstar, and Con Comp Industries. Most retail for $195, but considerable discounts are easy to find. A continuing decline in the cost of memory chips plus stiff competition has already brought the cost of the R.H. board down to $160. The last three companies are relative newcomers to the field and their prices are $130, $130, and $149, respectively.

Despite slight differences in looks and circuit design, all the cards perform basically the same function—they provide an additional 16K of RAM according to a standard control scheme that Apple established with the Language Card. All except one require that one chip be moved from the Apple motherboard and a jumper inserted in its place. For those who like lights, both the Microsoft and Computer Stop boards have LED indicators for RAM select, write enable, and bank select. These lights can be helpful when diagnosing certain problems, but in normal use they're hidden by the Apple's cover.

Except for price, the biggest difference among 16K RAM boards lies in their control over the F800-FFFF address space, normally occupied by the system Monitor. The Language Card, as you'll recall, contains an autostart ROM that takes control of this space regardless of what ROM is in the F8 socket on the motherboard. Most of the RAM board copies have elected to eliminate this ROM since it is not essential for RAM card operation (and the ROM is copyrighted by Apple). Besides, most Apples (all Apple II Pluses) already have an autostart ROM.

The Andromeda card has a switch that allows you to select either motherboard F8 ROM operation (presumably manual start) or RAM F8 (presumably autostart loaded in with Applesoft) on reset. The best solution, however, occurs on the R.H. Electronics board, which contains an empty ROM socket and four small dip switches. The switches select either RAM board F8 ROM, motherboard F8 ROM, or software selection between the two. Normal Apple ROMs or user programmable 2716 EPROMs may be used in the extra socket. If you like, you can add an outboard switch to select between ROMs.

Finally, there is a unique feature in the RAMEX 16 board from Omega Microware (actually, Omega is the distributor for the British company, Vergecourt Ltd.). Unlike all the rest, this board does not require any modification to the motherboard—that is, there's no IC to remove and no jumper—which makes the board easier to install and remove. Most people would not be willing to pay extra for this convenience, but, in fact, this is one of the *least* expensive RAM boards. Keeping the price down is quite a feat, considering that extra circuitry

ATARI

48K — w/disk

CRYSTALWARE

TRS-80

48 — w/disk

APPLE

48K — w/disk

THE FINEST IN FANTASY GAME SOFTWARE

At Crystal we are doing our best to provide the finest state-of-the-art graphic adventure software in the world. Our list of credits include the first indoor-outdoor graphic adventure, the first multi-disk graphic adventure, and now for the Atari, the first graphic adventure in the world which includes screen scrolling and animation. The era of the text adventure and games which are simple combinations of static graphics and text is rapidly drawing to a close. We attempt to utilize the full potential of your computer. True, many of our games use up to 48K and we only deal in disk products, but there are a lot of users out there who have worked hard to upgrade their systems to the max and we think they deserve games that will give their computer system a run for its money.

★★★ ADVENTURE GAMES ★★★

1-THE HOUSE OF USHER- Haunted house type adventure game with scrolling in the Atari version. Wander the creepy hallways of the three story castle based on Edgar Allen Poe's short story of the same name. Written in graphics, of course, with animation and sound. We have introduced a new mystery for another $100 prize. $29.95/1 disk

2-FANTASYLAND 2041 A.D.- The largest disk based adventure game in the world (that we know of). Enter the Hall of Heroes and prepare yourself for the greatest fantasy-role-playing game you will see for years to come. To win you must survive Congoland, Arabia, King Arthur, Captain Nemo, Olympus (a sea voyage), and Dante's Inferno (Hell itself). In both the Atari and Apple versions it takes up more than 400,000 bytes of memory and uses more than 400 hires screens. The winner of the contest described in the manual with this game will receive $1000.00 and a bronze trophy. We have pushed the award date forward to February 1982 to allow more people to participate in the contest. $59.95/6 disks

3-GLAMIS CASTLE- Yes, Pat and I are on our way to Britain to stay in the dreaded Glamis Castle. If we survive our real life adventure, we'll be measuring it and will be able to provide you with a 3-D game based on this ancient haunted site where King Duncan met his end at the hands of Macbeth. Our good friend, Mark Benioff, after much research, said there's a mystery room that has never been found in this castle and a half beast, half-man creature that guards a treasure therein. Our stay will be covered by the British media and we hope to share our experience with you through the writing of this game. $49.95/2 disks

4-BENEATH THE PYRAMIDS- You are an archaeologist in 1932 and must find your way through the perilous chambers beneath the pyramids to discover a golden statue of the cat goddess Bast. This game is in hires graphics, includes sound, your little man actually moves through the corridors which you can see on the screen. The monsters are animated and very agressive. There is a new $100 prize for the first to solve the mystery; which is a toughie! $29.95/1 disk

★★★ SPACE GAMES ★★★

5-GALACTIC QUEST- An excellent combination of Star Trek and Space Trader. Battle the animated Vegan fighters as you warp from galaxy to galaxy. At the same time, you may land on and trade with hundreds of planets. Super hires graphics and lots of sound. This has been one of our most popular games. $29.95/1 disk

6-SANDS OF MARS- Take an exciting voyage to the planet Mars via the Starship Herman. This game compared to the rest, is second only to Fantasyland 2041 A.D. It includes scrolling on the Atari and hundreds of full screen graphics. You can move your little man through the terrain of Mars; if, of course, you survive the exciting journey to Mars, which occupies the whole first disk. There is a new mystery and another $100 prize just waiting for some clever adventurer out there. Good luck! $39.95/2 disks

★★★ WAR GAMES ★★★

7-WORLD WAR III- You Atari gamers will have to see this in the Atari version to believe it! If your tired of war games which take 15 minutes a move and have a manual the size of a telephone book; but still want a complex, real-time action war game-this is it! It is designed for two arm-chair generals which may manuever up to 128 seperate type of units at a time. The game displays a map of Iran & Iraq in the first scenario and later on you will find yourself moving nuclear submarines and battleships through two world wars. This is not a boring copy of a board based game but an original war game which takes a lot of skill and may take weeks to play. $29.95/1 disk

8-WATERLOO II- If you had been Napolean would you have done a few things differently? Well as you approach this final battle you are equipped with the same forces, face the same opposition, and survey the same terrain which he did. We have done a great deal of research to make this historically accurate as well as extremely complex. Even the angle of sight, fatigue of the individual soldier, and his psychological profile are included in the calculations. Oh by the way, your opposition is no slouch. You may find it more difficult to change the course of history than you think! $49.95/2 disks

★★★ ARCADIA ★★★

9-LASAR WARS- Hires-3d space war simulation. Protect the earth from alien invaders. $29.95

10-LITTLE CRYSTAL- The first of our line of education software, which will be completed by December. It includes a very fine version of Hangman, Mr. Music; which transforms the computer into a piano, Gunk-a hilarious shoot-em up game, and Storytime- an anthology of bedtime stories featuring Herman, the cat, Oscar, the Hamster, and of course, Little Crystal. $39.95

11-IMPERIAL WALKER- A fine game pack written by our Atari programmer, Michael (graphics) Potter. Includes the Walker animation which is superb, Gunfight, and Lasar Nim, a game of 'how many robots'. $29.95

VISA

12-ADVENTURE PACK- (#1-4) $112

13-SPACEOUT- (#5-6) $58

14-THE WARRIOR- (#7-8) $64

15-ARCADE- (#9-11) $60

master charge

Crystal has many other fine fantasy and space games. For a copy of CRYSTAL VISION which includes a complete catalog please send $3.00 to the address below.

Our order lines are open 24 hrs. a day 7 days a week.

(408) 778-2966 CRYSTAL COMPUTER 17120 Monterey Rd., Morgan Hill, CA 95037

for adding address strobes and refresh for the RAM chips had to be added to the board.

Beyond 16K. It was almost inevitable that someone would continue to expand the amount of RAM available to the Apple II. All that was needed was some software to take advantage of the extra memory. The first attempts involved using more than one 16K board. When it became obvious that this could produce such useful results as relocating DOS to free up 10K of motherboard RAM, or enabling superfast program and data storage, people began to look for larger memory boards. Saturn Systems introduced their 32K RAM board and some interesting software. Then, as if we couldn't have predicted it, Legend Industries came out with a 64K RAM board.

Saturn Systems 32K RAM Board. For less than the price of most 16K RAM boards, Saturn Systems offers a new card with twice the memory capacity. However, what really makes this card a bargain is the extensive software that accompanies it. So before you can ask, "Now what am I going to do with that extra memory?" Saturn Systems offers you:

Movedos. This program allows you to put DOS in one half of the 32K board, freeing up 10K of RAM on the motherboard for program or data storage.

2. *Ramex.* A set of utilities that lets Basic make use of the extra memory on the RAM board.

3. *Disk Emulator.* Makes one or more RAM boards appear to the system as a *very* fast disk drive, using standard DOS commands.

4. *VisiCalc Expansion.* Lets *VisiCalc* take advantage of any extra memory in the system.

Note that the last three applications refer to multiple RAM board systems. Indeed, these programs will handle as many 16K or 32K RAM boards as you may have. The relocated DOS from the first program also has provisions for using an Applesoft/Integer ROM card in any slot. As you can see, the folks at Saturn Systems have been very busy creating a total system. Let's take a closer look at the hardware.

To make a 32K RAM board, you start with a 16K RAM board and then add eight more memory ICs. Obviously, this makes the 32K RAM board somewhat larger than a typical Apple peripheral board (such as a 16K RAM board). Installation is the same except that the memory chip is pulled from the second row on the motherboard instead of the top row. Because a switch protrudes from the back, the board must be installed in slot 0, 2, or 4.

When the board is installed in slot 0, the first 16K bank of RAM looks just like any other 16K RAM board. Thus it is compatible with Pascal, Basics, CP/M, and the rest while still offering an additional 16K of memory (note however that all present software relates to DOS/Basic use; Pascal and CP/M will not use the second 16K bank.)

A word of caution about power supply drain and heat generation: Like many other boards designed for the Apple, the 32K board exceeds the maximum current recommended to be drawn by any single peripheral (mainly on the +12-volt supply). What this means depends entirely upon your individual Apple and the number and type of other boards you may be using. You should also consider the heat generated by the board as well as the reduced air flow that it may cause. If you use more than one RAM board, you may find it wise to install a fan to keep the Apple operating reliably.

The 32K RAM board has a switch to enable either RAM or ROM after a reset. With the switch down, a reset will send control to the motherboard ROMs, including whichever F8 Monitor ROM is there. What happens when the switch is up depends upon whether the 32K board is write protected or not. If it is not write protected, then the reset will cause the RAM board to be selected; if it is write protected, then there will be no change in RAM/ROM selection and the reset will be processed by whichever Monitor is in control at that time. For typical use with Basic, what all this boils down to is this: Apple II Plus users will see no effect of the switch since there is an autostart monitor ROM on the motherboard and a copy of the autostart monitor is loaded into the RAM board along with Integer Basic. Apple II users will have manual start operation with Integer Basic; from Applesoft (on the RAM board), switch up gives autostart reset while switch down will take you to the manual start Monitor after pressing reset.

Movedos. One way to take advantage of the extra memory on the 32K RAM board is to move DOS onto the board. This is accomplished by the program *Movedos,* which also allows you to put an Applesoft/Integer ROM card into a slot other than zero.

Thus, it's possible to have both Integer and Applesoft in ROM, DOS up on the RAM board, and still have 20K of RAM left on the RAM board for other uses. If you think all this is easy, consider the fact that everything just mentioned must reside at the same address ($D000—$FFFF).

Thus, a scheme of bank switching is used; and, since Basic and DOS need to operate at the same time, a dynamic bank-select routine is required that must be transparent to the user. *Movedos* does this by placing the routine in the highest available RAM ($BE00—$BFFF, after DOS has been moved), thus freeing up an extra 10K of RAM on the motherboard. The significance of this extra motherboard RAM is that it is directly accessible and thus available for Basic program and/or data storage.

Similar programs that will move DOS onto other RAM cards have surfaced, but most are incomplete. Some require a complicated booting process, do not allow disks to be initialized with the new DOS, or leave the file buffers back in the motherboard RAM. *Movedos* suffers none of these problems and also offers a unique solution to bloading into the $D000—$FFFF memory range: you can configure the DOS to load into either bank of the 32K board or to another RAM board in a different slot. Careful examination of the *Movedos* program (a detailed, technical explanation included in the manual is very

Weather Strip
Exec SDS
November
1981

helpful) reveals a carefully researched, well written, and thoroughly documented piece of software. Saturn has even included a modification program that will fix Apple's *FID*, *Muffin*, and *BOOT13* utilities to work with the new DOS.

Ramex. *Ramex* (not to be confused with the RAMEX 16 board from Omega Microware) is designed to expand the amount of RAM available for use by Basic. Even though you may have one or more RAM boards in your machine, the standard Apple operating system and Basic interpreters cannot use this extra memory (except for holding the nonresident language or DOS). *Ramex* allows you to save, load, or run short subroutines or complete programs from this extra RAM memory. All *Ramex* commands are executed from Basic by setting up a command message string, CM$, and then doing a CALL 10. The command message consists of a command number followed by a set of parameters. One of these parameters can be the name of a variable to which the status of any *Ramex* command will be returned (zero if the command was successful; any other number indicates what type of error occurred).

Here is a list of the *Ramex* commands and their functions:

0 = no operation
1 = store complete program in extended RAM
2 = load and run program from extended RAM
3 = store subroutine in extended RAM
4 = load subroutine from extended RAM
5 = store contents of an array in extended RAM
6 = load contents of an array from extended RAM
7 = exchange contents of an array between regular RAM and extended RAM
8 = delete extended RAM segment (frees up space)
9 = connect new bank of extended RAM
10 = disconnect bank of extended RAM
11 = find total amount of free space in extended RAM
12 = get name of next segment
13 = (not implemented yet) load Basic program from disk into extended RAM
14 = initialize *Ramex*

Ramex can help you write new programs faster. Let's say you have a good input routine in one of your other programs. Another has a fantastic sorting routine and a third has nice output formatting. You wish to combine pieces of code from each program without having to key them into the computer again. *Ramex* to the rescue.

After initializing *Ramex* and loading the first program, you move the routine of interest into extended RAM. Say you want to save lines 1000-1500 of the present program in memory. The following command will store the selected lines in extended RAM under the segment "name" 1:

```
CM$="3,1,ERR,1000,1500":CALL 10
```

What this means is to perform function 3 (store subroutine) into segment number 1, returning an error code into the variable *ERR*. Anything stored in extended RAM is identified by a segment name. Segment names in *Ramex* are actually numbers in the range of 1-32767. After executing the given statement, the variable ERR will have a value indicating the status of the attempted operation, for example:

0 = operation was completed successfully
7 = selected segment already exists
9 = illegal segment name (unlike DOS, you must delete a segment before you can write to it again)
10 = no program area specified (did you try to save a range of line numbers that does not exist?)
12 = not enough room in extended RAM (sort of like *disk full*)

After saving the first routine, you would load in the second program and save the desired lines from this program into another segment. Finally, you would do the same for the third program (or just delete those lines you didn't want). Then, with just two commands to *Ramex*, you could bring back the first two segments stored in extended RAM. These would get added together and presto! you have all the desired routines already entered into your new program which is now off to a running start. Of course, like the Applesoft append utility, any line numbers loaded in that match existing line numbers will overwrite the previous line. Also note that *Ramex* works with both Integer Basic and Applesoft.

Disk Emulator. The disk emulator software was not available for review. The purpose of this software is to make DOS think that the extra memory is a disk drive. Thus data can be transferred to the RAM board with the usual DOS commands. Since there is no motor to get up to speed or seek delays, for example, transfer speed is *much* faster. Imagine typing *catalog* and seeing the directory displayed instantly on the screen. We will have more to say about this next month when we describe the Legend 64K RAM board.

VisiCalc *Memory Expansion.* Many businesspeople have run into the limitations of the Apple's memory capacity when using *VisiCalc.* With this new software, you can add 32K or more of extra RAM (that's already more than double what you have without a RAM board). This is the only program that does not come with the board; it sells for $100.

Summing It Up. If you're in the market for a RAM board, you should consider the 16K boards first. The primary factor here will probably be price since most boards are functionally identical. The switchable operation of the Andromeda and R.H. boards, plus the ROM socket on the latter, make them worthy of extra consideration. For ease of installation, the RAMEX 16 board wins hands down. You might also consider warranties and serviceability as factors in your decision.

If you would like to go beyond 16K of extra memory, then the Saturn Systems 32K board is an excellent choice. With all the software that comes with it, it is truly a bargain. But this is not the end. Next month, we will review the 64K board from Legend Industries along with some additional software. Then you will *really* have a tough choice to make.

MARKETALK News

Unless otherwise noted, all products can be assumed to run on the Apple II, Apple II Plus, and Apple III in the emulator mode and to require 48K and one disk drive. The requirement for ROM Applesoft can be met by RAM Applesoft in a language card.

□ *Joyport* from **Sirius Software** (Sacramento, CA) is a game port expander that lets you use four paddles (with all buttons functional) or one or two joysticks. Two connectors accept joysticks and afford fast response without delay between reads. Total compatibility with existing software and hardware, Sirius states—even with a shift key to TTL input #3 modification. Under $100. □ You're alone in a space outpost. Suddenly you're attacked from all sides by enemy fighters. Survival depends on your skill in directing propulsion jets against enemy ships and torpedoes and in strategic placement of shields to protect your base. *Outpost*, also from Sirius and written by Tom McWilliams, is a game that promises plenty of frenzied action. Eight levels of play; user-defined keys; sound on/off toggle; escape for temporary game interruptions; and restart function. Either DOS. Joyport, joystick, or keyboard. $29.95. □ Another far-from-serious Sirius game, *Beer Run*, finds you at the bottom of the Sirius Building, trying to catch Artesians. Ladders, elevators, and escalators help you ascend to other floors, but Bouncers try to catch you, too, while Guzzlers attempt to push you off ledges. Artesians like to throw cans of Oly off the roof; catching them gives you extra points. You may even sample the brew, but beware of imbibing so much that you lose your balance! A blimp trip from one building to another adds to the fun. Features include several play levels; escape feature for temporary halting of game; sound on/off toggle; color hi-res graphics. Author Mark Turmell. Assembly language, either DOS. $34.95. Joyport, joystick, or keyboard. □ **TypeShare** (Downey, CA) accepts sequential ASCII files from your Apple with modem and returns typeset copy, ready for paste-up and printing, according to your coding and specifications. First line in each file must be a line-count number for the text in the file; second line must contain type style and other instructions for the TypeShare center. Turnaround must not be critical, since typesetting is returned by mail or common carrier. $6 per foot of typeset copy on eight-inch paper. □ *MMS II* from **Computer Data Services** (Amherst, NH) is a memory management system for the Apple II that adds 10,700 memory bytes so you can run or write larger programs, create larger files and arrays, process more data internally to reduce disk access time, and increase the speed of your programs by reducing housekeeping functions. All 16K expansion boards are compatible with *MMS II*. Includes other features: copyable so it can become the *Hello* program on your work disk; automatically loads 16K RAM board in slot 0 in about two seconds; once initialized, *MMS II* will automatically run whatever program you previously directed it to; compatible with programs that use page 3 of memory. If you need both Applesoft and Integer Basic and simultaneously wish to have *MMS II* active, you'll need another 16K RAM or ROM card. *MMS II* will recognize the second card in whatever slot you select. Either DOS. $49.95, $2 shipping. □ **Computer Data Services** also offers *Andromeda*, a ROM board with utility ROM to allow automatic line numbering, program list controlling with page mode, restoration of a crashed program in memory, disk catalog alphabetizing, and disk creation without DOS. $125. □ *Saturn Systems* 32K RAM Expanson Board, also from CDS, lets you run Pascal, Fortran, Pilot, and other languages available for Apple II; increases *VisiCalc* memory by an additional 9K. Compatible with Microsoft CP/M and Z80 Softcard. $239. CP/M and Z-80 Softcard. $239.

□ The *MicroSpeed Language System* from **Applied Analytics** (Upper Marlboro, MD) marries the language Forth to the Intel 8231A Arithmetic Processor and an interactive compiler to provide enhanced speed and programming capabilities. Features include print formatting, hi-res plotting, turtlegraphics, and extended math functions. Two configurations are available—one using the 2 mHz version of the processor, the other the newer 4 mHz model. Both configurations include two text editors and 6502 assembler. *MicroSpeed II*, $495; *MicroSpeed II+*, $645. Manual—available separately—$35.

□ Physics students or engineers can learn physical properties on flow of gases or liquids for tube data engineering with a

menu driven flow calculation program called *Liquid and Gas Flow Calculation* from **American Avicultural Art and Science** (Saint Louis, MO). Touch key selection system to help nonprogrammers. Applesoft/controller card, 32K, either DOS. $40.

☐ **Micro Data Base Systems** (Lafayette, IN) announces the **Z65 Data Base Processor**, consisting of both hardware and software for the Apple: a Z80 CPU, 6502-Z80 software interface, and Z80 implementation of MDBS. Supports both host language (Applesoft) interface and query interface, allowing ad hoc data retrieval with a single, English-like statement. Also designed for extensive control over data redundancy, automatic data integrity and security enforcement, automatic transaction logging and recovery, and dynamic data base restructuring. Z65 package with Z80 CPU card, available through a single user license, $1,600; without card, $1,500.

☐ You can customize your disk catalogs with *Directory Master* from **Micro-Sparc** (Lincoln, MA) by creating attractive disk headers and catalog titles containing normal, inverse, flashing, and/or control characters. Any number of file names may be sorted alphabetically or reordered. You can also recover files accidentally deleted; write all changes directly to disk being modified, eliminating need for initting new disk or creating superfluous Poke files. ROM Applesoft, 1 or 2 disk drives, separate version for each DOS. $39.95.

☐ *Your Own Computer*, by Mitchell Waite and Michael Pardee, is a nuts-and-bolts book from **Howard W. Sams** (Indianapolis, IN). Discusses computer applications for home, office, and classroom; explains computer concepts; contains a glossary of terms. $7.95. Sams also offers a book called *Experiments in Artificial Intelligence for Small Computers*, by John Krutch, which presents programs in Microsoft's Level II Basic that can be converted to most Basic dialects. Game programs; problem solving. $8.95. *Sams Microcomputer Dictionary*, second edition, also now available. Includes such terms as Ada language, compilers, silicon-on-sapphire; also some noncomputer words like optical cable and laser annealing. Illustrated. Charles J. Sippl, author. $15.95.

☐ *Webster's Microcomputer Buyer's Guide*, by Tony Webster (**Hayden Book Company**, Rochelle Park, NJ), deals with such subjects as microcomputer theory and applications; independent software products; microcomputers and microcomputer systems; and CRT displays, printers, and terminals. More than one hundred buying sources listed. $25.

☐ Aliens hurl exploding balls from the top of your computer screen in *Blister Ball* from **Creative Computing** (Morristown, NJ). You defend yourself from a laser base as the number of bouncing balls increases, and the walls close in. The reliable laser base is also used in *Mad Bomber*, Creative Computing's second new release. Bombs dropped from a large space ship hovering overhead can be zapped away with a single direct hit from the laser. Paddles. $24.95 each.

☐ *Fast Spooler* from **Southern Semiconductor** (Norcross, GA) has been developed to save waiting time during printout, especially in small business applications. Capable of storing large amounts of printed material in its own internal memory to free host computer for other functions. Can be connected by either RS-232 serial interface or Centronics parallel interface. Spooler stands alone with own power supply, cabinet, and necessary software. Has its own 64K buffer. $695.

☐ Can the Confederation of Free Space ever enjoy a moment of triumph as it resists the domination of the Galactic Empire, conqueror of the entire solar system? The answer may lie in your intelligence and skill as commander of combat ship *Shadow Hawk I*—the confederation's last weapon against the imperial fleet. Challenge in *Shadow Hawk I* from **Horizon Simulations** (Medford, OR) is to prey on empire merchant ships, while evading enemy attack, in order to capture material needed to rebuild the confederation's armament. Triple-axis hi-res three-dimensional color graphics permit craft to roll, veer, and attack from any direction, so forecasting enemy moves is impossible, no matter how many times you play the game. Nine levels of accomplishment are rewarded by rank. Requires joystick. $49.95.

☐ The *Synergy-Card* from **Spies Laboratories** (Lawndale, CA) is a 16K RAM card with options for adding a parallel printer port (Centronics and Epson standard), a serial port (RS-232C), two interval timers, a real-time clock/calendar, and a BSR X-10 interface to your Apple. That's six slots' worth in one slot. Compatible with Pascal, Fortran, *VisiCalc*, and others. Card and manual, $195. Option 1 adds parallel port, serial port, and two interval timers with software (on disk) for $59. Option 2 requires Option 1 and adds real-time clock/calendar; $49. Option 3 requires Options 1 and 2, along with BSR X-10 interface for $39.

☐ **Automated Simulations** (Mountain View, CA) has added *Sorcerer of Siva* to its Epyx line of role-playing games. As a wizard with a variety of magic spells, you must battle an evil sorcerer and a multitude of monsters while searching through more than 300 chambers and five levels of a mine for treasures—and safe exit—as quickly as possible. Some treasures may help you; others carry the sorcerer's curse. Eight levels of skill. $29.95.

☐ **Syntauri Ltd.** (Palo Alto, CA) has a new operating system for its *alphaSyntauri* polyphonic real-time synthesizer. Among the new features are microtonal capability, user control of the frequency offset between percussion and primary waves, choice of linear or exponential curves for envelopes, and a vibrato package. Any player-designed waveform can be used as a low-frequency oscillator for the purpose of vibrato modulation; rate and depth of vibrato are controlled with the game paddles. The octave can now be subdivided into anywhere from one to thirty-one notes, and the frequency offset between primary and percussion waves can be varied in sixteenth-tone increments from a unison to an octave. Three additional utilities are available: *Auto-Pulse* offers automatic pulse wave generation, *B-3 Wavemaker* makes the alphaSyntauri emulate a Hammond B-3, and *New Plus* provides instantaneous change of instrument definition to facilitate live performance.

The utilities, by Steve Leonard, keyboard player for the rock group Cretones, are $50 apiece or $100 for the set; *Alpha Plus*, $35. Demonstration record for *alphaSyntauri* features selections from Bach to contemporary jazz to tropical rain forest sounds. $2.

☐ **Inmac** (Santa Clara, CA) introduces transfer switches that control up to six peripherals to maximize Apple system use. You can increase the number of terminals, printers, and modems connected to a single CPU without the expense of rewiring or the inconvenience of changing or juggling cable connections. Twenty-five-line RS232 cable transfer switches, $175 to $260; two-port t-switches for nine-line and fifteen-line RS232 cables, thirty-seven-line RS449 cables, and equipment with fifty-pin cables/connectors, $165 and $225.

☐ Modem users can join in discussions via the Conference Tree, a network of separate bulletin boards. Open conferences you may join currently, along with computer access numbers, include: "Civil Liberties, Space, Lasers, Amateur Radio, Robotics, (415) 928-0641; "Computer Communications, War, and Peace," (415) 526-7733; "Forth Computer Language," (415) 538-3580; "Berkeley Local—Everyone Welcome," 8 p.m. to 8 a.m., (415) 548-4683; "Telecommunications: Hardware, Software, and Modems," (415) 381-3668; "Forth and General Micros," (213) 394-1505; and "Health Information for the Disabled," 6 p.m., to 9 a.m., (415) 325-7066. Unless otherwise noted, conferences are usually available around the clock. Participation is free. In case you'd like to start your own Conference Tree, **Communitree Group** (San Francisco, CA) offers software, manual with operating instructions and general telecommunications information, start-up conference, and publicity for your conferences. $120.

☐ *Paymaster Payroll System* from **Masterworks Software** (Lomita, CA) can process up to one hundred employees in any combination of payroll periods from daily to yearly. Among *Paymaster*'s features are access to all data elements on file; ability to print any screen as well as standard reports and programs; accommodation for tax law changes. Compatible preprinted checks available. Any standard printer; 2 to 4 disk drives. $275.

☐ *Micro-DSS/Finance*, financial modeling software package from **Ferox Microsystems** (Arlington, VA) now has virtual matrix capability with an expanded work area of 32,000 cells to accommodate large modeling tasks. Comes with full-scale report generation, command files, color graphics. Pascal, 64K, 2 disk drives. $1,500.

☐ *Memory Master 1.0* from **Legend Industries** (Pontiac, MI) is a memory management extender equipped with Legend 64KC card of almost any 16K RAM card similar in function. *Memory Master* gives an additional 8.5K of RAM memory on the motherboard; each disk catalog displays the number of unused sectors; machine language programs can access DOS RWTS routines through standard DOS page 3 vectors without using additional page 3 space; user can flip between 3.3 and 3.2 without rebooting; .STAT command promptly displays current DOS version in use; .BSTAT command displays hexadecimal starting address and length of last binary file either bloaded or BRUN. $34.95.

☐ If you're a **CompuServe Information Network** (Columbus, OH) subscriber and have questions about using your Apple, want to exchange software information, find a user group near you, or air a gripe about any of the above, you can use a new information/advice service called *The Micro Adviser*. It's being put together by **Battery Lane Information Services** (Bethesda, MD), publishers of the *Computer Consultant* newsletter and directory. Future plans for TMA include a list of micro consultants and contract programmers, hardware reviews (in addition to the software reviews already available through TMA), and book reviews. Subscriber suggestions are welcome. CompuServe subscription fee for home use is $5 per hour, 6 p.m. to 5 a.m., plus connect fee.

☐ **Cross Educational Software** (Ruston, LA) offers a series of general physics study disks. A few of the following programs use calculus, and nearly all have hi-res graphics: *Vectors and Graphing*, $10; *Statics*, $12; *Motion*, $12; *Conservation Laws*, $10; *Circular Motion*, $15; *Thermodynamics*, $20; *Electricity*, $12; *Optics*, $20; *Atomic Physics*, $30; *Solar System Astronomy*, $30; and *Stellar Astronomy*, $30. The last two are less technical and could be useful in a physical science survey course. Company also has a series of Christian education programs that include *The Christmas Story*, $12; *The Methodist Story, Bible Books*, and *Hymn Book*, $15; *Presbyterian Child's Catechism, Bible Books*, and *Hymn Book*, $15; *Caroling*, $15; and *The Night Before Christmas/Popular Christmas Songs*, illustrated with lo-res moving pictures, $15. Games from Cross are *Hangman*, containing a child's riddle game and a famous sayings game for adults, $10; children's *Dinosaurs*, $15; *Blitzkrieg 2*, a target game with ten skill levels and fast, noisy action, $15; and *Aquarium*, $25.

☐ *The Disk Labeller* from **Practical Software** (Pomona, NY) automatically prints labels for disks showing sectors free, sectors used, and DOS on disk, just by pressing the return key. Features include built-in default functions, choice of printer slots and label sizes (four-inch or five-inch.) Special labelling sections enable you to print duplicating labels and labels for disks that can't be catalogued. You can also preview catalog so that labels can be printed with titles, numbered or unnumbered files, and disk numbers. Package includes two program disks and one hundred five-inch labels. $42.95.

☐ The second edition of *Computer Quarterback* football simulation game is out from **SSI** (Mountain View, CA). There are still three variations—pro, semi-pro, and solitaire—but now the player draft has been expanded to all three. Players start with $3 million to buy the teams they wish, or they can buy a separate disk and choose from all the 1980 pro teams. There are also new visual and audio effects such as whistles at the end of each play and an elaborate display after each touchdown. Comes with rule book and player aid cards. Written by Dan Bunten. ROM Applesoft. $39.95. ☐ **SSI** has also introduced two historic battle simulations by Practical Design Group: *The*

Battle of Shiloh, based on a major Civil War battle, and *The Battle of the Bulge: Tigers in the Snow,* which simulates the famous World War II German offensive of December, 1944. Rule book and player aid cards included for both games. ROM Applesoft. $39.95 each.

☐ **Evolution Software** (Toronto, Canada) offers *Mail Mate,* a mail merge system that can operate with *Magic Window* word processor or by itself as a mailing/phone list system. To merge letters produced from *Magic Window,* you must produce a soft copy in the printer subsystem. (Soft copy not restricted in size.) *Mail Mate* then accesses the soft copy and merges it with the addresses and names selected from your address list. A few highlights of the system: string search on names of uncertain spelling; ten selection fields; flexible specification of selection codes for printing and logical ANDing between selection fields; can operate with single disk drive holding up to 500 addresses per disk; prints address list with either one or two addresses across; field for area code and phone number, 6502 machine language or Basic, one or two disk drives, either DOS, $68, or $85 Canadian. ☐ Evolution also offers *VMIX,* a *VisiCalc* file utility program for consolidating up to fifty files into one. Consolidated file contains totals and averages of any of the input file fields. You specify labels and equations used in consolidated file, and they needn't be the same as those in the input files. 6502 machine or Basic, 1 or 2 disk drives, either DOS. $76, or $95 Canadian.

☐ *Sight 'N Sound* from **Compugraphics** (Saint Louis, MO) lets you transform sound into action on your hi-res screen. After loading program from disk, sound is generated by playing a tape on a cassette player connected to your Apple. Program monitors changes in frequency and plots to screen accordingly. For example—in the demo called LIPS—as you talk into the microphone or play a taped voice, the lips move in synchronization. Disk includes several demos with options for creating and editing your own designs as well as using designs provided. Y connector supplied for directing sound to both Apple and speaker; no hardware modifications necessary. Written by Ray Balbes. ROM Applesoft, either DOS. $29.95.

☐ *Aircraft ID* from **Rainbow Computing** (Northridge, CA) helps airplane buffs draw and identify more than forty aircraft from World War II to the present in hi-res three-dimensional graphics. You can also play Name That Plane with one or two players, or run a slide show of aircraft drawings. $19.95.

☐ *Ms.Speller II,* available from **Intelligent Computer Systems** (Arlington, VA), will process as input any Apple Pascal text file and output processed text to another Pascal text file. A dictionary/proofreader, it can help you with spelling and allow you to create your own dictionary of frequently used technical terms, proper names or foreign words. Text formatter included. $75.

☐ *Release 5.30* from **Microsoft** (Bellevue, WA), designed for the firm's Basic Compiler, now implements CHAIN with COMMON as an aid to menu driven and turnkey systems programming. Using CHAIN, you can develop a large system of programs to communicate through a COMMON data region. *Release 5.30* is available for CP/M systems including Apple II with Microsoft SoftCard. $395.

☐ *The Count* is a blackjack system from **Multi Data Service** (Ashland, OR) that plays blackjack with you; also teaches correct card play and simple, effective card count system. Either DOS. $24.95. ☐ *The Liberator,* also from Multi Data, is a subroutine library system providing a warehouse of usable subroutines documented and set up for immediate exec'ing to any program while letting you modify provided subroutines and add personal favorites to the system. More than fifty routines include a calendar function, dollar formatting, decimal rounding, hex converter, disk access variables, and line print tab field routines. Either DOS. $29.95 ☐ Multi Data's *Manipulator,* a text file utility system, allows you to change, add, modify, and print text files in any fashion you wish. Included are routines for field switching and file size changing; sort systems; and line print formatting for any kind of hard copy. Either DOS. $34.95.

WHY THE MICROSOFT RAMCARD™ MAKES OUR SOFTCARD™ AN EVEN BETTER IDEA.

Memory — you never seem to have quite enough of it.

But if you're one of the thousands of Apple owners using the SoftCard, there's an economical new way to expand your memory dramatically.

16K ON A PLUG-IN CARD.

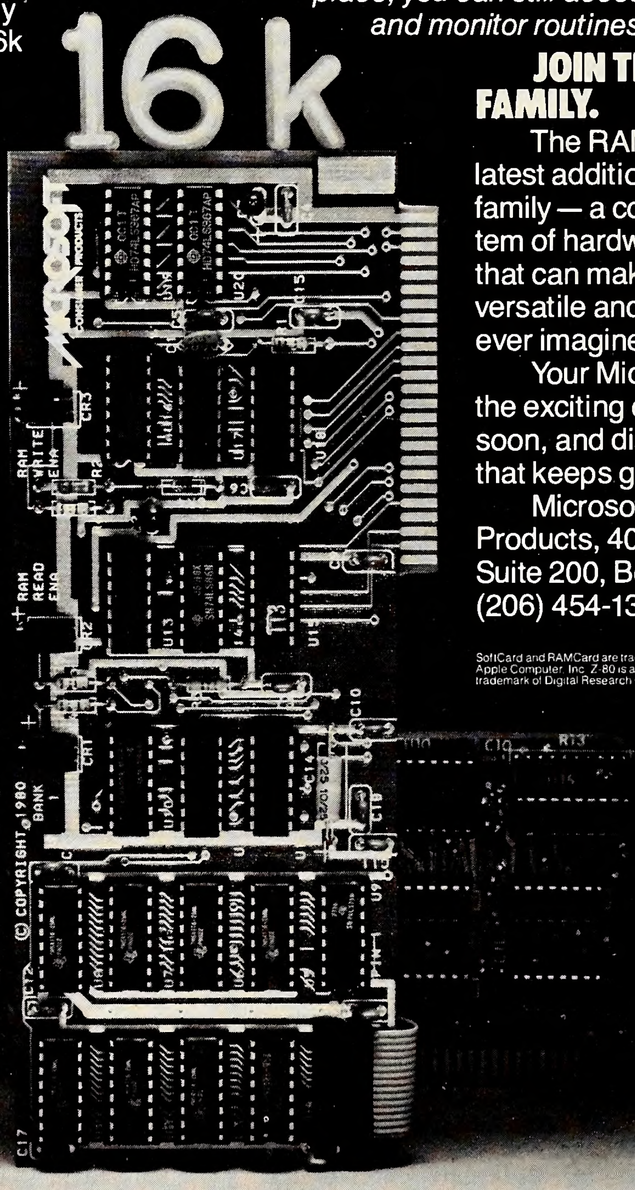
Microsoft's new RAMCard simply plugs into your Apple II,® and adds 16k bytes of dependable, buffered read/write storage.

Together with the SoftCard, the RAMCard gives you a 56k CP/M® system that's big enough to take on all kinds of chores that would never fit before (until now, the only way to get this much memory was to have an Apple Language Card installed).

GREAT SOFTWARE: YOURS, OURS, OR THEIRS.

With the RAMCard and SoftCard, you can tackle large-scale business and scientific computing with our COBOL and FORTRAN languages. Or greatly increase the capability of CP/M applications like the Peachtree Software accounting systems. VisiCalc™ and other Apple software packages can take advantage of RAMCard too.

And RAMCard gives you the extra capacity to develop advanced programs of your own, using the SoftCard and CP/M. *Even with the RAMCard in place, you can still access your ROM BASIC and monitor routines.*



JOIN THE SOFTCARD FAMILY.

The RAMCard is just the latest addition to the SoftCard family — a comprehensive system of hardware and software that can make your Apple more versatile and powerful than you ever imagined.

Your Microsoft dealer has all the exciting details. Visit him soon, and discover a great idea that keeps getting better.

Microsoft Consumer Products, 400 108th Ave. N.E., Suite 200, Bellevue, WA 98004. (206) 454-1315.

MICROSOFT

☐ *Stock Forecasting System* from **Urban Aggregates** (Columbia, MD) provides specific buy and sell advice on stocks; recommendations based on computer evaluation of moving averages, price momentum, and risk cycles. Momentum gap analysis and relative stock strengths also shown. Fifteen menu driven programs with three ways to update files (including accessing Dow Jones data base) and two ways to create files. Written by Robert W. Lovell. Graphics tablet, modem, printer, and 2 disk drives needed to use all program features. $175, demo disk $9.95.

☐ *Chat* from **Lovell's** (Corpus Christi, TX) is designed to provide your Apple and Hayes Micromodem with the flexibility you need to exchange and process data from other computers. Allows two-way phone conversations that can be saved in 26K buffer or printed as they happen. Especially useful to talk with network systems such as CompuServe, Dow Jones, and The Source. Includes four programs, three text files, binary code, and manual. Requires Hayes Micromodem; printer optional. ROM Applesoft or language card. $40.

☐ *Disk Emulator* from **Legend Industries** (Pontiac, MI) simulates an almost instant access disk drive for your Apple. Every byte of RAM memory on the Legend 64KC card can be accessed with the standard DOS 3.3 disk commands in Basic or with the Apple RWTS subroutines in machine language. Uses only 512 bytes of memory but will support up to six Legend 64KC cards. $700. Disk Emulator 2.1 software package, $49.95.

☐ Legend also offers a memory management system for your Apple II equipped with DOS 3.3 and Legend 64KC card; also supports most 16K cards. *Memory Master 1.0* provides a full 44K bytes of program storage within the 48K motherboard by relocating Apple DOS into any one of the four 16K banks in the 64KC card. Also manages your Integer or Applesoft firmware card. Each disk catalog displays number of unused sectors on disk; (.Flip) command allows you to flip between 3.3 and 3.2 without rebooting. $34.95.

☐ *Depreciation Planner* can help you keep track of depreciable business assets for accounting and tax planning purposes. Incorporates both previous depreciation methods (to be used for assets purchased before January 1981), as well as new depreciation methods (to be used for assets purchased after January 1981). Faster than manual record keeping; reduces chances for error and alleviates excess paperwork. **Dakin5 Corporation** (Denver, CO) has designed the planner to work independently or interface with the firm's *Controller* or *Business Bookkeeping System*. Applesoft, 2 disk drives. $395.

☐ Heat—one of your Apple's worst enemies—can cause erratic operation and shorten component life. *Apple Cooler* from **Concepts and Systems** (Jacksonville, FL) contains a fan and replaces your Apple cover with industrial-quality plexiglass, leaving room for a nine-inch monitor and two disk drives, or a TV set with drives on top. $159.50.

☐ *Meet the Presidents* slide show/game presents portraits of thirty-nine United States presidents in full-color hi-res graphics portraits created by computer artist Saul Bernstein. Display routines gradually reveal each portrait; player is given clues and must type in the correct president's name. Music accompanies completed portraits; points are tallied for speed in identifying presidents. You can change game clues. From **Versa Computing** (Newbury Park, CA). $39.95. ☐ Versa Computing's *AXE (Advanced X-tended Editor)* provides you with a text editor-style extension to the standard Applesoft operating system. Operating with DOS, Monitor, and Applesoft, *AXE* remains transparent until you call on it by one of more than thirty commands. A few of its features are global search and replace; full character insert, gobble, and delete modes; full statement insert and delete; four list formats; auto line numbering; user programmable keyboard macros; and lowercase character entry. At any point, the resident Basic program may be run; *AXE* shuts down and resumes operation automatically when program halts for any reason. For Apple II or Apple II Plus. Applesoft, 48K RAM. $69.95.

☐ **Scholastic Inc.** (New York, NY) has released its premiere issue of *Electronic Learning, The Magazine for Educators of*

the 80s. This bimonthly magazine is designed to be a central resource for teachers and administrators. Special introductory subscription rate for six issues, $15.

□ **Software Resources** (Cambridge, MA), announces *TrendSpotter*, a business graphics and forecasting analysis package. Generates color graphic displays, calculates and displays trend lines, performs mathematical and statistical computations, prints graphic and tabular data, and edits and updates files. Can generate and read *VisiCalc* compatible files. $175.

□ For the medical profession from **Charles Mann & Associates** (Yucca Valley, CA), a new publication called *Micro Medical Newsletter* is designed to provide advice on the use and selection of applications for use on microcomputers in medical offices. Current issues provide detailed reviews of accounting and insurance claim management systems available, as well as reviews of applications software. Sample issue is available free upon request. Year's subscription, $25. □ Also from Charles Mann is *The Reprogrammable Data Base Program*, which allows users with minimal programming knowledge to reprogram the system to meet their data needs. System includes detailed manual. Extended version available for the novice programmer includes a seventeen-lesson course on Applesoft. Also has a data base configuration program, a file creation and management element, sorting utility, report formatting utility, query and search element, and file transfer utility. Handles up to 9,999 records of up to 255 characters each, sorts a 1,000-name mailing list in three-and-a-half seconds. Allows programmer limitless modifications to make the system fit almost any business, educational, or personal need. Applesoft. $99.95. Program with two-disk, seventeen-lesson Basic course, $149.95.

□ **Battery Lane Publications** (Bethesda, MD) has a new guide for anyone wishing to sell programs they've written themselves. *Software Wanted: How and Where to Sell Your Program* is intended to bring software authors and marketers together. Several companies that market software are actively looking for programs written by independents, according to BLP. The book surveys software publishers and distributors to determine what kinds of programs they buy, what royalties they pay, and which ones offer help with programming and documentation. Information on more than sixty companies is provided, along with tips on what to look for before signing a contract. *Software Wanted* is being sold on a money-back guarantee basis for $25. Dealer inquiries invited.

□ **Hayden Book Company** (Rochelle Park, NJ) announces two new programs. *Inventory Management System for Stock Control* contains a complete and current stock overview. Manual part of the package is written for the novice. Sections include "Practice Run," to familiarize you with the system's features and functions, and "Reference Section," which provides detailed technical information on all of the system's features and functions, including a collection of usage suggestions. $175. □ *Dentistaid* is designed to streamline all major time-consuming tasks performed in the dental office. Menu oriented program will automatically print standard insurance forms, monthly statements, patient recall notices, daily summary of work performed and payments received, production analysis and display of individual accounts. Three disk drives, printer, Softcard. $11.00.

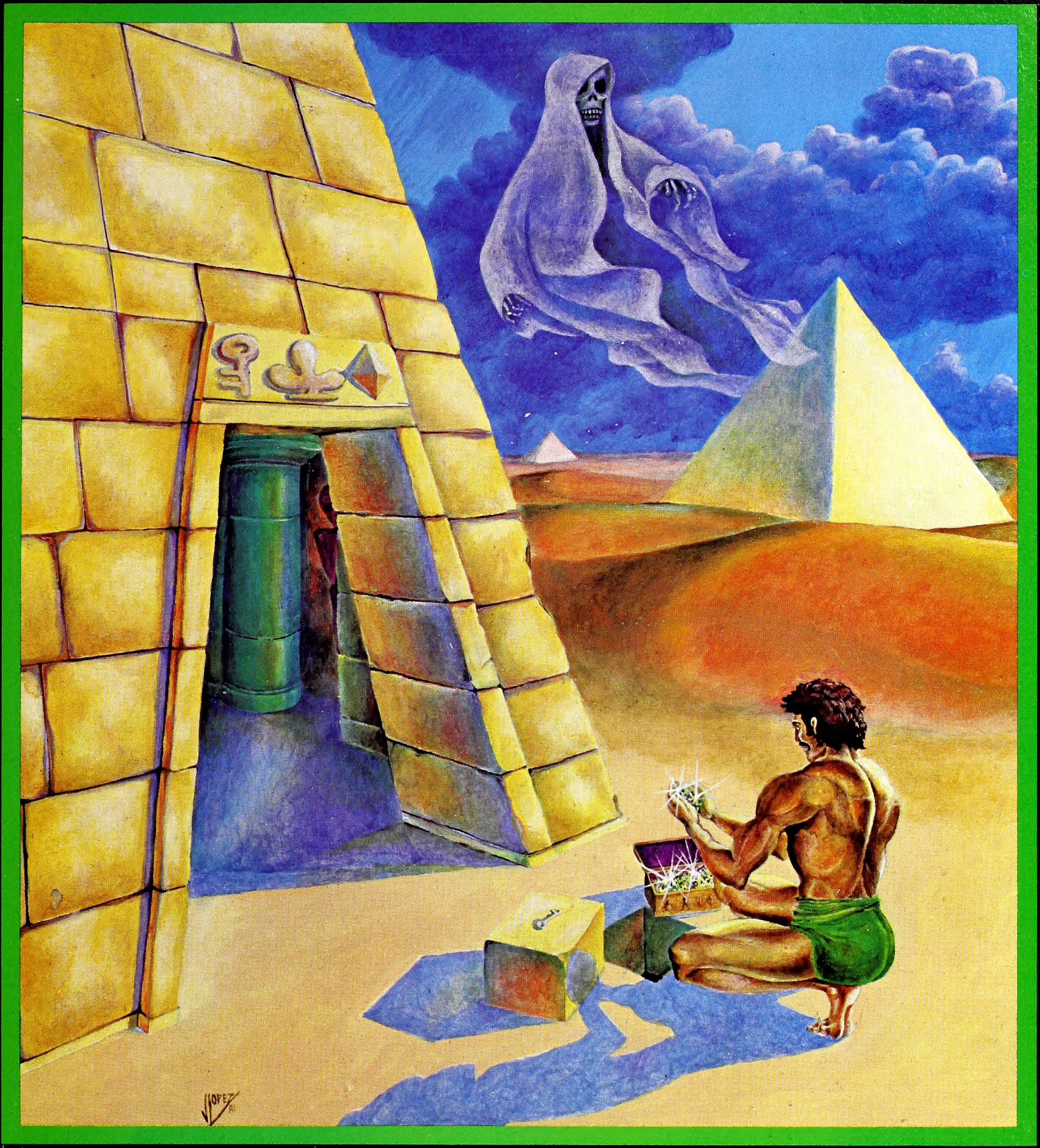
□ The fifth release in the adventure series *SwordThurst* has arrived from **CE Software** (Des Moines, IA). In *The Green Plague*, by Donald Brown, an epidemic is sweeping the nation with hundreds of green bodies dropping dead in the streets. Object is to save the kingdom and yourself, and time is very short. Adventure was created for the 1981 World Science Fiction Convention in Denver, Colorado, where the high score was 3,898 points. The first *SwordThrust* disk is required to use any of the other games in the series. ROM Applesoft. *SwordThrust* #1, $29.95; #2 through #5, $24.95.

□ **Charles Mann and Associates** (Yucca Valley, CA) has released another medical applications package—this one for the Apple III, called *Medical Office Management III*. Includes appointment management, private patient accounts receivables, and insurance form preparation. Designed for large medical practices; has cycle billing and time of service insurance compatible billing. Requires 128K, 1 to 4 SOS-compatible hard or mini-floppy disk drives. $1,595.

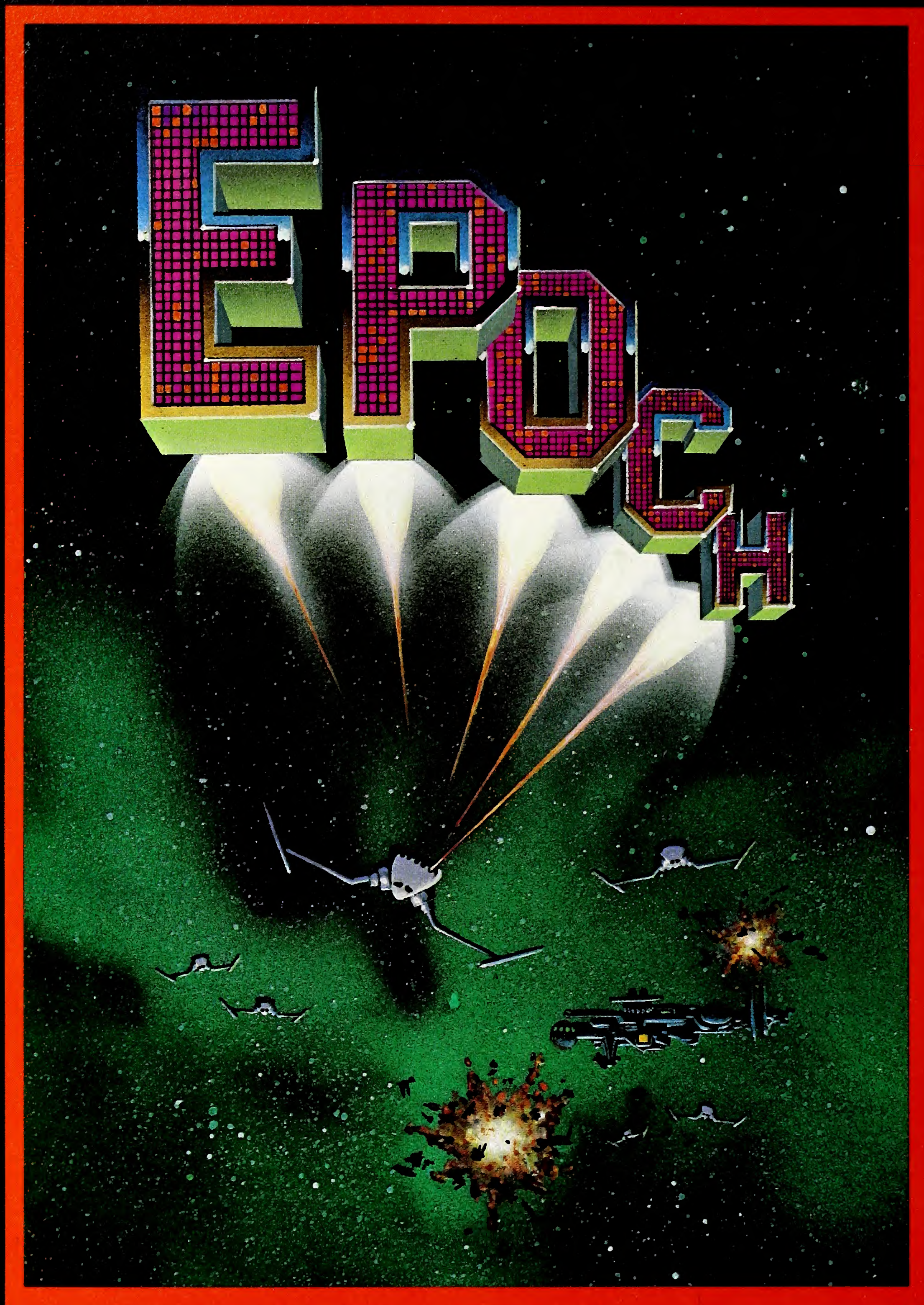
□ *International Microcomputer Dictionary* from **Sybex** (Berkeley, CA) is a pocket guide containing term definitions, acronyms with their pronunciations, and numbers used in microcomputer contexts, as well as a ten-language vocabulary of essential computer words. Another feature, called "The Numbers Game," lists electronic parts numbers commonly used in microcomputer parlance. Separate section gives more than two hundred computer terms with their equivalents in eleven foreign languages, followed by a standards/specs section and a section listing micro system/component suppliers. Paperback only. $3.95.

□ If you're conversant with APL language, you can use its problem-solving power on the Apple. **Vanguard Systems** (San Antonio, TX) offers the complete *APL/V80 Apple Software Package* with end-user license, object code disk, documentation, and special APL character generator. $500. Specifications are subject to change, but Vanguard has also announced its APL/Apple program with RAM, $675; with Z80 Softcard, $850; or with both Softcard and RAM, $995.

□ *Asort* from **C and D Software** (Little Rock, AR) is a machine language utility that adds new capabilities to in-memory sorting. You can sort up to twenty-five one-dimensional Applesoft arrays with seven keys in any combination of ascending or descending order while maintaining the parallel association of array elements. Ten thousand integers sort in less than forty seconds, according to C and D. Manual has instructions for saving tape. No minimum memory requirement; program can run on as little as 16K. ROM Applesoft. $30. ■

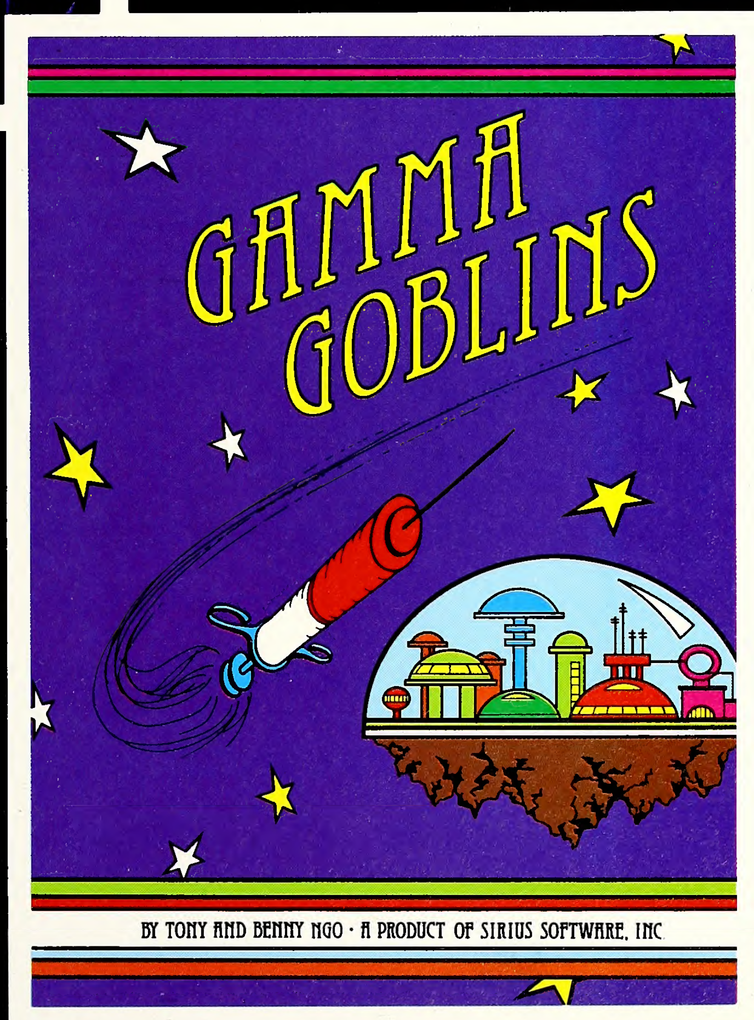


Play Copts & Robbers
In The Tombs Of Ancient Egypt

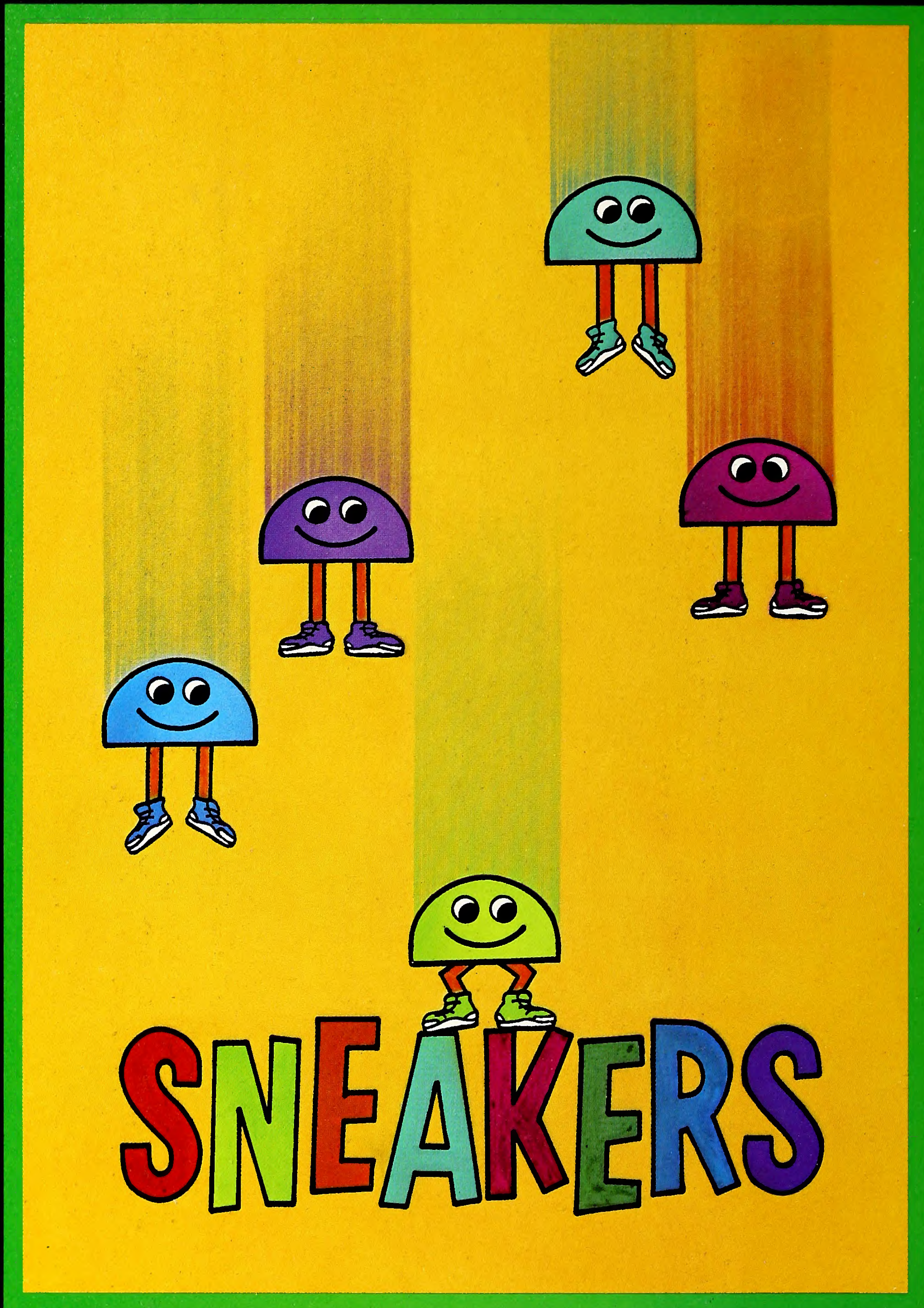


Where your secret weapon is the fourth dimension...

In the center of an orbiting space station you are protected only by a revolving force shield. Enemy forces are advancing from all directions and begin to place killer satellites in orbit around your station. And then, look out for the meteors!



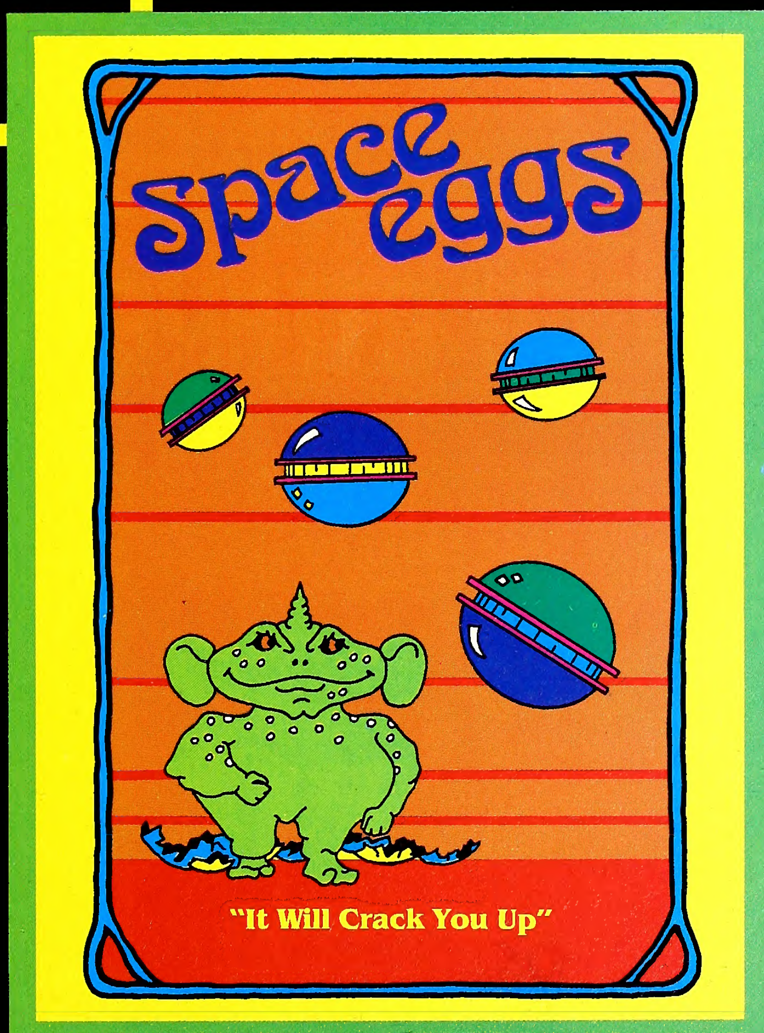
A "bloody" good game for the true-blue game freak. Your mission in this exploratory operation is to deliver whole blood to Hemophilia, a city in the sky, and return to Anemia Base before the Gamma Goblins overcome you. A real heart stopper!



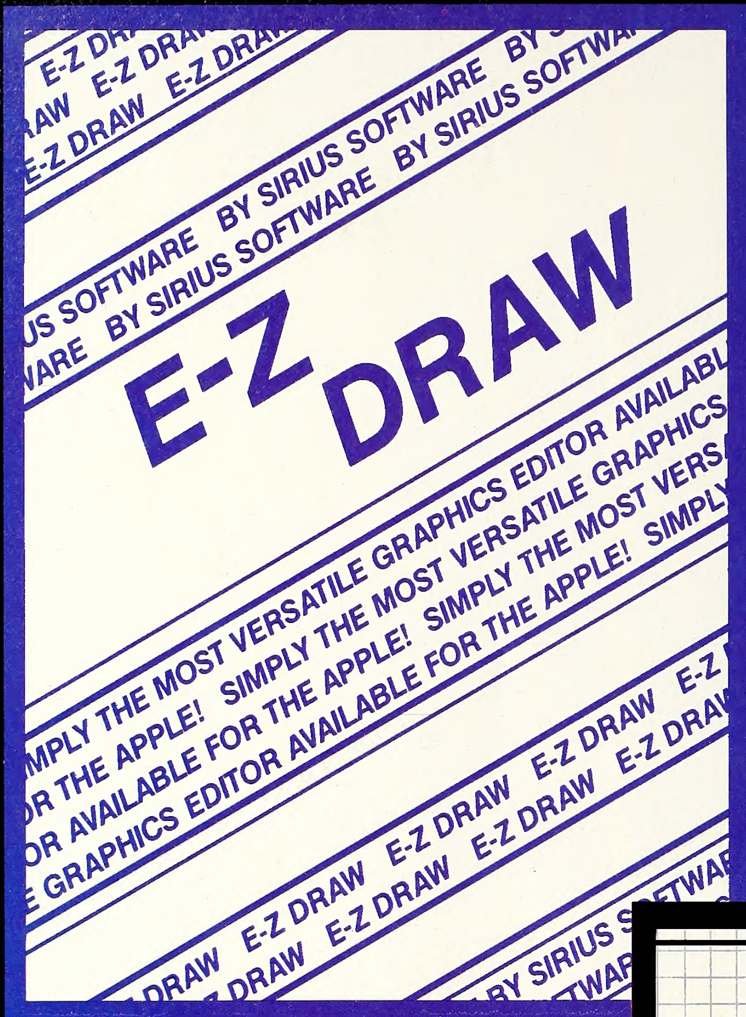
What say we go out and stomp a few???

Endless Excitement Stomping Sneakers And A Swarm Of Other Creatures

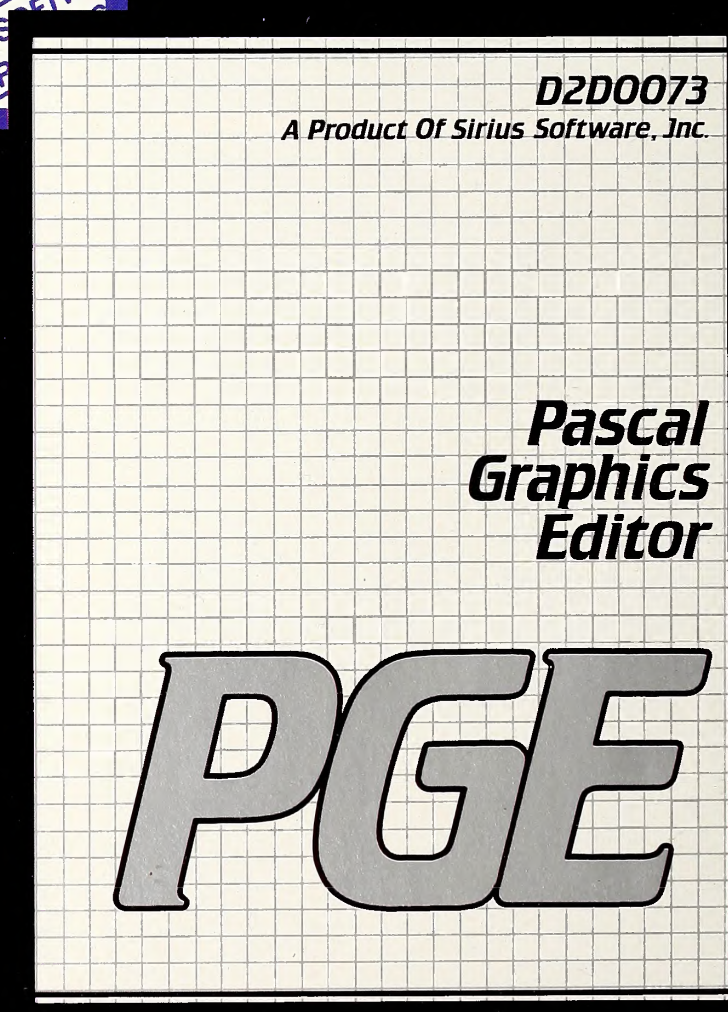
Phantoms Five simulates a fighter-bomber mission in real time, three dimensional color graphics. While you try to make your bombing run, you have to avoid being hit by anti-aircraft fire and fight off enemy aircraft as well.



Hatch some fun with the Spiders, Wolves, Lips, and Fuzzballs. Space Eggs will crack you up! Each package includes a multi-color T-shirt iron-on that says "I FRIED THE SPACE EGGS."



This is the graphics editing package we based our business on. Includes the Higher Text Character Generator by Ron & Darrell Aldrich and over 20 original and imaginative type styles.



The professional graphics editing package for use within the Pascal environment.

GORGON is here...

THE EARTH HAS ENTERED A TIME WARP . . . AND THE BATTLE HAS JUST BEGUN

BOTH BARRELS

A two game pack featuring "High Noon" and "Duck Hunt" You'll love the bad guy that falls off the roof and the dogs fighting over the ducks. Fun for the young and the young at heart.
Copyright 1980 By Sirius Software, Inc.

STAR CRUISER

Save yourself from the swooping aliens! This is a fast action arcade style game that can be played from ages three and up, but beware, the difficulty increases with each new wave of aliens.
Copyright 1980 By Sirius Software, Inc.

CYBER STRIKE

Interstellar challenge for the dedicated arcade gamer. You are in command of a light transport ship equipped with Hyperspace Drive, Antimatter Torpedoes, Local and Galactic Sensors, Meteor Shields, and an Instrument Panel which continually tabulates all information vital to your mission. You alone can prevent the clone take over of the allied settlement bases. WARNING . . . this game requires practice to play successfully.
Copyright 1980 By Sirius Software, Inc.

AUTOBAHN

Hair raising excitement at 120, 160, and 200 kilometers per hour! Drive through heavy traffic, oil slicks, narrow roads, and dark tunnels (with headlights). Watch out for the fire trucks! Only on the Autobahn can you drive this fast.
Copyright 1981 By Sirius Software, Inc.

PULSAR II

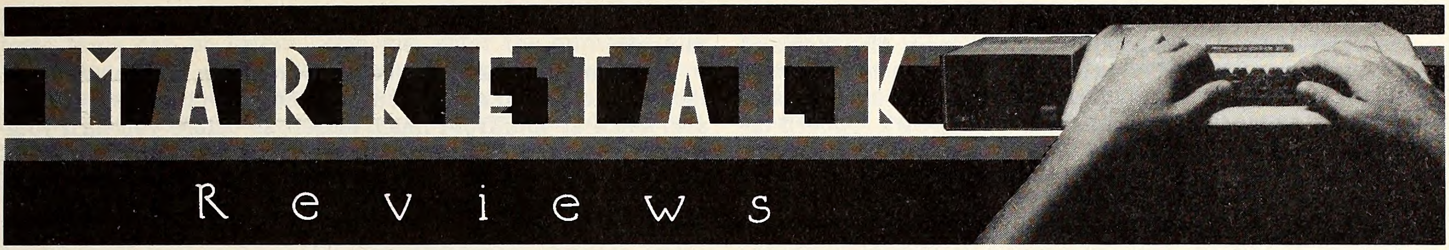
A unique two game series that provides scoring options for separate or combination game play. To destroy the "Pulsar" is no easy task. It is surrounded by spinning shields that send out orbs of energy aimed directly at you. "The Wormwall" places you in one of the strangest mazes ever created. The walls do not connect. Openings only occur temporarily as moving colored segments in the walls cross. In addition, there are munching mouthers in each level of the maze ready to gobble you up should you misjudge the time and location an opening will occur.
Copyright 1981 By Sirius Software, Inc.

Contact Your Local Computer Dealer For More Information • Dealer Inquiries Invited

Sirius Software, Inc.
2011 Arden Way #2, Sacramento, California 95825

PROGRAMMING: Cops & Robbers was programmed by Alan Merrell and Eric Knopp. Epoch was programmed by Larry Miller. Orbitron was programmed by Eric Knopp. Gamma Goblins was programmed by Tony and Benny Ngo. E-Z Draw was programmed by Nasir Gebelli and Jerry W. Jewell. Pascal Graphics Editor was programmed by Ernie Brock. Sneakers was programmed by Mark Turmell. Gorgon, Phantoms Five, Space Eggs, Both Barrels, Star Cruiser, Cyber Strike, Autobahn, and Pulsar II were programmed by Nasir.

SYSTEM REQUIREMENTS: All software mentioned in this advertisement require an Apple II or II+ with 48K with the following exceptions: E-Z Draw requires a 48K Apple with Applesoft in ROM (or a 64K Apple II or II+) Pascal Graphics Editor requires an Apple II or II+ with Language System.



Unless otherwise noted, all products can be assumed to run on the Apple II, Apple II Plus, and Apple III in the emulator mode and to require 48K and one disk drive. The requirement for ROM Applesoft can be met by RAM Applesoft in a language card.

Cyborg. By Michael Berlyn. Do you enjoy science fiction? Do you like getting thoroughly involved in a good story? Do you delight in a complicated plot, especially when all the pieces come together in a remarkable but totally logical conclusion?

If you can answer yes to any of these questions, you're apt to be as excited as we are about this new adventure from science-fiction author turned adventure programmer Michael Berlyn. Filled with puzzles and enigma, all of which have logical solutions, *Cyborg* makes you the protagnoist in a well plotted short story. As in David Mulloch's *Prisoner*, part of your quest is to figure out what's going on. Unlike *Prisoner*, where learning what was going on was the means to winning the game, in *Cyborg* discovering the plot merely leads you to identifying what your purpose is and, consequently, what you must solve to ensure a happy ending—and a win.

The premise is fascinating. The time is several centuries in the future, and you are a person in whom an extremely advanced computer has been implanted; to wit, a cyborg. This puts you in the uncomfortable position of having two brains in one body, both of which must thrive if either is to survive. As the story begins, not all your logic circuits are working, and your human memory is suffering for some reason also. Your two minds can converse and discuss, however, and help each other solve the confusion they're suffering.

If this is a device, it is so well integrated into the story that you won't notice it as such. Nevertheless, as a device, it is the perfect means to make natural and meaningful the normally unnatural exchange between you—the Apple user playing an adventure—and the character the programmer uses to be your senses in the adventure. It is as though you and a friend are making this trip together, not as though you are sitting in an armchair commanding a legman to do your running around. It becomes natural to ask for help, since it's not the programmer who answers, thus destroying the illusion of the game; it's your computer brain, your alter ego, your subconscious.

In fact, much of the description of place seems to come from the traditional omniscient narrator we never notice but just accept in fiction. When the computer mind of the cyborg speaks, its words are prefaced by an asterisk. Dialog you input appears after a *greater than* sign. Before long, you'll easily accept the place descriptions as what you're seeing, not what you're being told about. You can always confer with your computer half and get its comments on your location or on some object you've discovered.

When *Cyborg* begins, you are in a very strange forest that seems to go on forever. Because of your disorientation, which you share with your computer brain, you are aware only that you need power for your computer side and food for your human body. As you wander around, trying to make something of it all, you come upon strange phenomena, none of which seems to add up.

Occasionally, you may find temporary power sources and food. In fact, you must find these to go on. So as it would be to any starving person, your intent, despite your quandary, is to find sustenance. By the time you have some, you'll have stumbled on enough potential sources of information that you'll know how to begin seeking answers concerning your whereabouts and purpose. You'll also be thoroughly hooked.

Some traditional adventure commands are a little different in *Cyborg*: they've been altered to fit the situation and to add to the milieu. Instead of looking at things, you scan them (it's even grammatical, as are all commands in *Cyborg*). You scan yourself to discern your energy supply, your damage, your apparel, and your supplies. In other words, no inventory. Commands such as "go door" will make no sense to your computer side; but "enter cabinet" will. Articles are cast to the winds, but syntax is not.

As you progress in *Cyborg* and your situation and purpose begin to take form, you'll become increasingly absorbed, even beginning to sense an urgency (that power supply is crucial). You're apt to be as caught up in this adventure as you might be in a good book.

Cyborg is a text adventure; but carefully integrated into the plot—well hidden and essential to winning—is a brief action skill game in color graphics. Because what you must do to succeed at this game inset is so crucial to the forwarding of the plot, this graphic interlude works. Just be sure to save your game before you attempt it!

The best hint for solving *Cyborg*, which is no hint at all, is merely that nothing in this game is illogical or random. You are always in control. It's a game of the computer adventure genre that fulfills all the best of the genre: in playing it, your wits are challenged and success appropriately gives you the reward of delighting in your own mind's competence; and, as a tool for developing efficient mental and logical processes, it is unsurpassed.

As a small point, and a finishing touch, the author comes across as having great respect for his audience. All aspects of the program are finished with courtesy and consideration.

With all this, in its introduction of true plotting and realistic accommodation of the requirements of the genre, *Cyborg* introduces the most exciting advances in adventuring since the original *Adventure* began the whole wonderful thing. MCT

Cyborg, by Michael Berlyn, Sentient Software (Denver, CO). $32.95.

Genetic Drift. By Scott Schram. The name implies evolution, and that's what this game is all about—sort of. It's an arcade-type game, a shoot-'em-up, but your shots, instead of killing, simply cause your targets to evolve to the next stage in their evolution.

This is a theme that you would expect could go on forever—assuming *you* could avoid being wiped out. In fact, this is one of very few games of the arcade genre that has an end, a point to achieve at which you absolutely win. You even get a visual prize.

What do you suppose evolves? At first, your targets are mere shapes. They evolve to different colors and different shapes and finally into a familiar shape: television sets. Your gun sits center screen aiming out at four sets of four targets each. If you can evolve all four of any set into television sets, they stop shooting destructive missiles at you and begin sending valentines to you instead. These bring points; but it's very hard to remember not to shoot at any white shape coming at your gun. If you erroneously shoot a heart, the targets revert to objects two evolutionary stages earlier than those you began with the first time, and you must let them evolve all over again.

If you figure that you can train yourself to ignore a side that's sending hearts and thus avoid shooting one by accident, think again. Occasionally, a set of TV sets gets confused and sends an upside-down heart. Unless you shoot it and thus set it straight, the entire set of TVs turns into those early—hostile—mutants.

Each level of the game reflects one stage of evolution. On the third level and above, four satellites join the enemy. They also shoot at you, from much closer, so before you turn your regular foes into TV sets, you must get rid of the satellites. They too have several levels of evolution, though. And the higher you go, the faster and more frequently the objects and satellites shoot at you.

Besides giving you a philosophical point to ponder (what does it mean that everything evolves into television sets? Would that be good? Then why do the TV sets send love messages?), *Genetic Drift* gives you a whole new pattern on which to increase your hand-eye coordination. The G, Y, H, and space bar control your aim and you shoot with escape—one shot at a time—or A—four shots, one in each direction, at a time.

Overall, Broderbund has another winner here, hard, complex, and laced with plenty of variation.

As a bit of fun on the side, Broderbund is offering a Broderbund T-shirt to the first winner of *Genetic Drift* in each state of the U.S. If you win, write—*don't* phone—Broderbund. MCT

Genetic Drift, by Steve Schram, Broderbund (San Rafael, CA) $29.95.

Diet Analysis. By Javed Aslam, M.D./Tess Enterprises. By tracking food intake over a thirty-day period on *Diet Analysis*, you can discover the combinations of food intake and activity that result in gaining, losing, or maintaining your weight. With this utility, you can chart the caloric, carbohydrate, and protein content of nearly seven hundred foods.

Central to the program is the food composition information. The task of sticking to a high protein, low calorie, or low carbohydrate diet is likely to be much easier if you are aware of the nutritional makeup of your favorite foods.

Before you begin to use the program in earnest, you'll be asked to enter such identifying information as your name, sex, height, current weight, size of frame and level of activity, as well as your date of birth and the current date. After analyzing this information, the program tells you whether your weight is within the ideal range for a person with the characteristics you have specified. If so, you are congratulated; if not, editorial commentary is dispensed with and only the facts—ideal weight, percentage of overweight or underweight—are given.

You can set up files for as many as ten different people, making this program ideal for a family or a group of friends to use. Once you've opened a personal file, you can begin to enter your food intake for the day. The process of food entry is well-prompted and easy to master.

The program allows you to store thirty-days' worth of food intake information. Then it begins to recycle, making room for additional daily information and storing fifteen-day average data. Because the program sets up a round file, information is available to you at all times about your most recent thirty-days' worth of food entries. Fifteen-day averages are stored as long as there is room on disk for them.

A data analysis routine allows you to compare and contrast the nutritional composition of your daily intake with the daily requirements for a person of your physical characteristics, compare your food intake over a five-day period, or take stock of your fifteen-day averages.

Another nice feature of the program is the casual analysis option. If you're contemplating the protein content of a six-ounce bar of semisweet chocolate, this menu option will give you the facts before the fact, rather than after. This information may help you get your cravings to more manageable levels (at least it will let your consider the gravity of your nutritional sins before you commit them).

A utilities option gives you the ability to delete files if you really want to, but the program itself makes it virtually impossible to delete a file accidentally.

Also part of the utilities option are the quick-mode and recipe features. Quick mode can be used when you get fed up with entering the same foods over and over. Once you have set things up, just enter breakfast, and the program will refer you to the list of breakfast foods you've established. Your

customary egg, bacon, toast, coffee, will be there (up to eighteen items can be listed); now just enter the numbers and quantities of the appropriate foods. Using the recipe option, you can enter the ingredients in Uncle Roger's hunters' stew, specify the number of servings you have prepared, and the number of servings you've eaten. Then, just relax and let the program do the figuring for you.

A major strength of this program is its delightfully clear, logically organized instructions. Even inexperienced computerists won't be in danger of losing confidence, patience, or interest before they figure out how to use this program. The written documentation is also well done, but the prompting and clarity of the program itself makes the separate documentation almost unnecessary. JEV

Diet Analysis, by Javed Aslam, M.D., Apple Computer/Special Delivery Software (Cupertino, CA), DOS 3.3. $45.00.

Personal Finance Manager. By Jeffrey Gold. Special Delivery Software's contribution to the plethora of family finance accounting systems has much to recommend it and a few shortcomings as well. The program handles up to two hundred entries a month from a maximum of fourteen separate accounts—a checking account, a cash account, and twelve credit cards. Entries may be classified by budget category; a total of twenty-four budget categories is allowed. The budget module lets you specify projected totals for each category for each month of the year, and subsequently displays percent variance between actual and projected amounts. There's also a lores graphics feature that shows the bad news pictorially, using color to highlight that part of the bar graph that rises above your budgetary expectations.

One of the strongest features of the program is its search/sort/edit routine. You can seek out and display all entries for a given month or range of months, or you can limit the search according to various keys—particular budget categories, particular payees, specific check numbers, deposits only, cash only, specified credit accounts, tax-flagged only, and so on. The range of choices is impressive. You can even use several of these search keys in conjunction with one another to flush out, for example, all tax-flagged items in the months of May and June, payable to anyone with a first name of George and attributable to the budget categories of food, transportation, or entertainment. The one search criterion that *PFM* lacks that might be desirable is transaction amount; it will not hunt down all checks between fifty and a hundred dollars, for example.

Once a search is ordered, the program offers you the additional option of sorting the results on any one of four keys—check number, date, amount, or budget category.

Once the results are displayed, you may, if you wish, edit or delete individual entries. If an item is shown to be reconciled with the bank and you change its amount, that item returns to unreconciled status.

Reconciliation is another area where *PFM* scores high. The program has a trial reconciliation feature that steps you through your data entry by entry, letting you tick off those that have shown up at the bank. When you're done, if the numbers don't add up right, you don't have to undo anything; the program cleans your slate and lets you start again. On the other hand, if the trial reconciliation is successful, you can opt to make it permanent.

PFM is well documented and easy to use. Default replies are provided for most of what the program asks of you, reducing keystrokes. The legal moves available at any given point in the program are shown at the bottom of the screen. The prompting even goes so far as to tell you when you're about to respond to a get statement, rather than an input, so that you don't hit unnecessary returns and find yourself on the second line of the ensuing module.

The biggest failing of *PFM* is its lack of a dedicated print routine. You can hit control-P at any time and print the current screen, but that's not really satisfactory. For example, if you just wanted a printed list of your checks for a given month, you could do no better than send them out a dozen at a time, complete with column headings and prompts for each group. Furthermore you'd waste at least half your paper, since the screen-dump approach only uses forty columns.

Other shortcomings are the limit of twenty-four budget categories and the fact that you can't set up budgets for different sources of income. In fact you can't even budget aggregate income. Nor will the program record check memos. Withdrawals from a checking account that are not in the form of a check—like cash subtracted from deposit of a paycheck or withdrawals from an automatic teller—have to be entered under a phony check number like 9999; the program won't let you code them as what they are.

PFM comes with a backup copy. Master and backup are uncopyable, except for the data file that you create. This can be transfered with *FID* so your records are safe in the event of a disk crash. CS

Personal Finance Manager by Jeffrey Gold, Software Dimensions Inc., Apple Computer/Special Delivery Software, Cupertino, CA. ROM Applesoft. $75.

Bug Attack. By Jim Nitchals.

"The ants go marching one-by-one, hurrah, hurrah;
"The ants go marching one-by-one, hurrah, hurrah;
"The ants go marching one-by-one,
"The little one tries to chase the sun,
"And they all go on to. . . ."

Oh, hi. You caught me singing along with *Bug Attack*, the new arcade game by Jim Nitchals—you know, the fellow who wrote *Star Thief* and *Asteroid Field.*

What? You've never heard of sing-along computer games? Well, that's your loss. No, that's not all it does. It's an action game. The music accompanies the ants marching through the clover.

What do you mean, "What kind of threat is an ant in clover?" You just haven't seen the daggers they're dropping at me . . . . That's right, daggers—well, microdaggers. Who

cares if real ants can't wield daggers? If it's real you want, go sit on an ant hill.

Huh? Oh, sure, get me a soda, too.

That was quick. You came back because you thought you heard the computer playing the *1812 Overture.* Well, you did. It plays it at the end of the cartoon. That's a reward for making it through three levels. No, you don't do anything during it. You just watch it. The animated graphics in the whole program are terrific. Bright colors in a bouncy style that makes you smile a lot. No, I won't tell you what happens; you'll have to make it that far yourself.

It's a worm. Yes, you're right, a worm isn't technically a bug. It might as well be, though. It's blue because blue worms are pretty. Blue worms make you feel good. So do orange ants. If it's reality you want, go dig in the garden.

You think it looks like fun. Well, you're right there. No, you can't have a turn as soon as I get killed. Maybe I won't get killed. Ever think of that? No! I'm reviewing it; I *have* to play it over and over. Who says I'm addicted? It's my job. It's work. Sure, I think you'll enjoy it, but go buy your own *Bug Attack.*

They're swarming medflies. They're all back because the last one got my insect spray. Can't you keep still for a minute?

Okay. I got them. You can talk now. The ants are a piece of cake. When I get to the next round, you'll see the millipede. When you shoot at it, it speeds up—a lot. If you get really high up, there's still another bug.

What was that? It was an explosion, obviously. Haven't you ever heard an explosion before? Geesh. What got me? Why should you care? I might've made a new high if you hadn't distracted me with your questions. Why is it so important what got me? Go sting a medfly!

I got killed, what does it matter now?

Oh, for Pete's sake.

An ant.

". . . The ants go marching two-by-two,

"The little one stops to tie its shoe,

"And they all go marching down

"To the earth

"To get out

"Of the rain." MCT

Bug Attack, by Jim Nitchals, Cavalier Software (San Diego, CA). $29.95.

Agenda Files. By Peter Meyer and Apple Computer. *Agenda Files* is a personal or small business memo system that enables you to create files for keeping track of what you need to do and where you need to be—today, tomorrow, the day after tomorrow, and in the future.

This program consists of six files. The first four (today, tomorrow, the day after tomorrow, the future) cycle and update automatically. As the day passes, tomorrow's tasks become today's, three days' advance notice becomes two, and so on. If you're in doubt about the day of the week or month on which an event falls, you can consult the calendar and view any month of any year from 1981 to 1999.

The other two files, called the first list and the second list, can be renamed to suit your own purposes. You can then use them for such things as making a list of projects to be done on a rainy day, or for recording phone numbers and addresses you don't want to lose track of. It's easy to clear out the contents of a file when you want to reuse it, so you might want to use one of these files for plotting out or keeping track of the steps involved in a major event or project.

A tutorial program on the master disk allows you to get your feet wet by making entries to sample files almost immediately. And unlike some other file management programs, *Agenda Files* does not require you to supply an initialized blank disk before you can start using the program.

The quick-entry option is another helpful feature. If you've already used the program once on a given day, you can use this option to add, delete, or transfer an item without running the entire program again. This quick-entry process should take only about thirty seconds from start to finish.

Most of these features are fairly standard ones for an agenda/file management program. As such programs go, *Agenda Files* is not terribly sophisticated. But then, not everyone requires a system with the capability of tracking accounts or distinguishing between appointments and general reminders (such as *Time Master,* Image Computer Products), and not everyone wants or needs to invest time or money in a more elaborate system.

What *Agenda Files* offer is a clear, uncomplicated, easy-to-get-the-hang-of program that's reasonably priced. Although it lacks some of the niceties that can be found in other more costly programs, it certainly holds true to its promise of helping you organize and uncomplicate your life.

A somewhat inconvenient aspect of *Agenda Files* is that although you can add, delete, move, and rearrange entries easily, simple, one-step editing of an entry is not possible. More sophisticated programs (for example, *Disk Calendar II* from Telephone Software Connection) do allow editing.

A reassuring feature of *Agenda Files* is the file fix option. If you should do something unmentionable like accidentally hitting reset or opening the door of the drive while the program's writing to the disk, file fix will notify you via a screen prompt and will do what it can to repair the damage.

Agenda Files comes with data disk, backup disk and instruction manual. A printer, while not required, is recommended. JEV

Agenda Files, by Peter Meyer. Special Delivery Software, Apple Computer, Cupertino, CA. $35.

The World's Greatest Blackjack Program. By Warren Irwin, Carl Cooper, and Lance Humble. Oh no, you say, not another blackjack program. Who would have the effrontery to lay that on us, and why would Special Delivery publish it?

Well, no, it's not just another blackjack program. It's a rather special one. It will teach a card-counting approach to the game that will—allegedly—give you an advantage of sev-

eral percentage points over the house at any casino in the world.

The theory goes like this. In blackjack, unlike craps or other so-called independent trials games, the odds are constantly shifting according to the composition of the deck remaining. If you can identify those times when the odds are good for you, you have the basis for a winning game. Just bet high when the deck is hot and low when it's not. Further improvement in your chances can be brought about by modifying your playing strategy—your decisions about whether to stand or take a hit in particular circumstances—according to the status of the deck.

The World's Greatest Blackjack Program teaches a fairly simple card counting technique that requires you to observe only two categories of card. Threes through sixes you count as plus one, tens and faces as minus one. You keep a running count based on these two categories, and the count correlated against the size of the remaining deck yields an index that determines your play. Sounds complicated and it does take some practice, but the program will teach you patiently as only a good computer program can.

You don't have to learn this card-counting technique to use the program. It will also teach you to play correct basic strategy, showing you the most advisable play in all situations assuming you're not counting cards.

But you don't even have to worry about basic strategy. You can just forget about education altogether and play blackjack. The game module itself is unusual. It's fast (it's written in Pascal, though you don't need Pascal to run it), and the hi-res graphics simulate real-life casino play in loving detail. Even the shuffling algorithm mimics the rituals observed in Las Vegas—with a certain variance from perfect interleaving built in for the sake of verisimilitude.

About the only difference from casino conditions, apart from the fact that the chips stay inside your machine, is that the program has room for only six players, not seven. You and your friends can fill as many of those seats as you like, and you can let the computer play the others; or you can just go head-to-head with the dealer and leave those other seats empty.

Part of what's nifty about this program, from the standpoint of the serious blackjack player, is that it contains a menu of rules options that cover all major variations in the game. You can choose Las Vegas Strip, where the dealer stands on soft seventeen, or downtown Las Vegas, where he does not; or you can play the rules that prevail in northern Nevada or in Atlantic City. Surrender, doubling after pair-splitting, resplitting of pairs, and second hits on split aces—all can be allowed or disallowed at your discretion. And you can choose to play with one, two, or four decks.

The program has three main modules: a card-counting trainer, a strategy trainer, and the game mode. In card-counting module the program just rolls cards past you, singly and in pairs, at a speed controlled by you. You count them and the computer checks your count.

The strategy trainer presents you with situations on a graduated scale of difficulty and asks you for the appropriate play. The program times your response and recycles the problems that you miss or that take you the longest to answer, so you're trained not only for correct response but also for speed. The strategy module will teach you either the basic (noncounting) or counting strategy, as you specify.

In the play mode, if you wish, you can tell the computer you're playing either the basic or counting strategy and it will beep if you make a wrong move. Unfortunately, it will not also verify that the size of your bet is appropriate. The correct bet size is not as clearcut a matter as the appropriate play so it would have been a touch awkward to program the computer to verify it, but from the standpoint of educational value it would have been nice to have this feature.

And, speaking of education, you can also, if you wish, put this program on automatic pilot and let the computer play all six hands against itself—with each hand using either card-counting or basic strategy. Just watching how the system performs over time can be a sobering experience, because, even if the odds favor the counting player on the long haul, on the shorter haul most amazing disasters can occur. Perhaps this form of education may be the most important value the program has to offer.

The World's Greatest Blackjack Program, by Warren Irwin, Carl Cooper, and Lance Humble, Apple Computer/Special Delivery Software, Cupertino, CA. $50.

SAT English 1 by Eileen Shapiro; **U.S. Constitution Tutor** by Myrna Helfand. These programs treat two subjects in the same excellent format for its purpose. Both programs are designed for high school students for review study in preparation for taking SAT exams, but each program could be used to learn a basic amount about its subject.

Opening menu gives a choice of several aspects of the subject. Having chosen one of these, you may choose either to be tutored or to be tested. If you choose test, you're given around twenty-five multiple-choice questions. You answer—with the option to change your answer or go on protecting against typos—and go to the next question. At the end, your score is revealed and you may choose to see the questions you missed.

If you choose tutoring, you're still given the multiple-choice questions, but the program informs you immediately whether you're right or wrong. If you're wrong, it explains why, removes that choice from the display, and asks you to take a second choice. On your third wrong answer, the program tells you the right answer and requires you to type the full answer before you can go on.

Categories in the English program are those used in the four verbal SATs: analogies, antonyms, sentence completion, and reading comprehension. The U.S. Constitution program is divided into two sections for beginners, two for intermediate learners, and two for advanced. MCT

U.S. Constitution Tutor, by Myrna Helfand, and *SAT English 1,* by Eileen Shapiro, Micro Lab, Highland Park, IL. $30 each.

Impressions

□ **The Inspector.** By Bill Sefton. For the Apple owner who is interested in what is in memory and on his disks and would like to know what would happen if he changed it, *The Inspector* can be more fun than any game and far more useful. With it, you can look at the contents and organization of any sector of any disk written with DOS 3.2 or 3.3, including the VTOC, directory, and track/sector lists. You can look at random access or sequential files to verify that they were written correctly. And you're not limited to just looking. You can write whatever you want into any sector you want. Warning: Writing to a disk can be dangerous to the health of your disk. Whether through ignorance or carelessness, an error can result in loss or damage to the information on your disk. If you're not sure of what you're doing, try it out on a back-up disk first.

With the ability to write to the disk, you can salvage most, if not all, of a blown disk, change the data in a data file, and delete DOS (if you do the latter, you'll lose the ability to boot on that disk but you'll add two full tracks of storage). You can read and write to both thirteen and sixteen sector disks. You can specify a new slot, drive, track, or sector at any time. You can specify the page of RAM memory you wish to fill with the data you read from the disk. You can automatically read successive sectors into the same memory buffer or into successive pages.

A memory map can be requested that will show which sectors on the disk are free and which are reserved. You can search a single sector or a full disk for a specified string. The

string may be as many as nine ASCII characters or $0F hex bytes. A search for the string, "APPLE," on the DOS 3.3 System Master disk took just under four minutes and found the string in forty different sectors.

Additional capabilities include the ability to dump all of memory, including ROM, to the screen with start, stop, fast, and slow scroll controls. You can search all memory for a specified string. This takes less than four seconds.

No matter what you have on the screen, you can request the display be either in hex format or in ASCII. Since the ASCII format will usually be a mix of normal, inverse and flashing characters which is very hard to read, you can request that all characters be shown as normal. Any information that can appear on the screen can also be printed. *The Inspector* permits editing memory on a byte-by-byte basis. If the display is in the ASCII format, you can specify that the characters you input are lower case (you need lower case capability for this), flashing, or normal (but not inverse). Both entering and exiting *The Inspector* are fast and easy and can be done without disturbing the program in memory at the time.

The Inspector is available either on a PROM chip or on a disk. Apple IIs and Apple II Pluses with access to Integer Basic in either ROM or RAM on either the motherboard or on a plug-in board can use the chip version. Apple II Plus without access to Integer Basic must have either a RAM card or a language system and requires the disk version. *The Inspector* is fast, flexible, and complete. One improvement that might well be incorporated in an update is the ability to step through the sectors of any specified disk file. All in all, *The Inspector* is a very useful utility. Omega, Chicago, IL., $49.95.

☐ **TG Joystick, TG Products** (Richardson, TX). When you play Sirius Software's *Epoch*, you may choose any setting for your joystick. You may have the screen scroll down when you move the stick up and right when you move left or you may have it scroll down when you move the stick down, and so on. You may also choose the level of fine tuning: will you have it respond fast and with vigor or more slowly but evenly? Whatever you do, this program is a great test of joystick.

The TG Joystick passes the test with ease. The joystick is of the automatic centering variety, but it's not married to the center—meaning that the aim stays easily with you. Using this joystick, you don't feel like you're waging a hopeless battle with the stick every time you want to leave the center. Two surface levers enable you to set the center where you want it; they have just enough tension to be easy to set but not easy to move accidentally in play.

The casing is rather large and heavy enough to sit securely but not heavy enough to be an encumbrance when held. It's the same color as the Apple. This stick is intended to be used sideways with the stick on the right and the buttons one above the other on the left. The buttons are plastic cushioned for comfort, but they are definitely buttons, not inset covers that can disappear inside the mechanism on a game that gets a bit violent. Buttons are far enough apart that they can't be confused in individual play, but they're close enough to be depressed simultaneously by using your whole thumb.

Ted Gillam has designed a superior accessory in the TG Joystick. $59.95.

☐ **Rings of Saturn** by Dan Minardi, Dakin5/Level-10 Software (Denver, CO). Level-10's first arcade-type game shows far more promise than their first role-player.

Rings of Saturn is nicely done in hi-res. As pilot of a ship whose mission is to rescue a sister ship's crew from a wreck near Saturn, you must fly through enemy craft and through the ice-ball filled outer rings of the large planet, rendezvous with the injured ship, and return through the same obstacles to the mothercraft.

Time, fuel, speed, and potential damage all come into play. An extra obstacle is the need to remember a code number without which you can't get everything working. It's not so hard to remember on the way out, but try remembering when you're set to come back! So write it down.

A nice feature is your repair robot. Control-any-function puts your robot to work fixing that part of the ship; it's very fast and easily may save your life.

You cannot control your course and the course never changes in *Rings of Saturn.* You'll always run into enemy ships at one point and into iceballs at another. However, there are five levels of difficulty, and the challenge to master these may make this game just the ticket for some computer gamers.

A very nice touch is the score record. High scores are recorded with players' initials. Both the record of high scores and the score of the last game played are saved and appear each time you boot the game. DOS 3.3. $39.95.

☐ **Space Quarks,** by Chris Jochumson, Broderbund Software (San Rafael, CA). Numerous tiny colorful shapes dance about the screen in Broderbund's latest arcader. Some are oblivious to you. When you shoot them from your tank along the bottom, they explode and disappear. Others however—of a different color but you never know which color—are not the least oblivious and will happily dive bomb you if you should mistakenly hit them.

Each time you make it through one full bunch, you gain a tank and find yourself faced with another assemblage of quarky creatures in another configuration. When you've mastered getting through the first half dozen or so groups, the game goes to disk for a new bunch. At least we believe it's a bunch. We've not made it through the first set after the load. In this group, all are the same color, so most of your ability to tell which are sitting ducks and which kamikazes is gone.

This game takes a while to grow on you; then, it's a real challenge. We believe there's more to it than we've been able to discover yet. If you go farther, let us know what you find. $29.95.

☐ **Derby**, by Jean Valentin, BiTaction (Saint Louis, MO). Horse races are apparently one of the most difficult sports to simulate well on the computer. *Derby* is probably the best attempt to appear yet.

Immediately in its favor is its use of real horses in random races. Valentin apparently knows a bit about racing and has chosen great horses of past and present, including some aficionados consider great even though they aren't household names. Examples would be horses who never won the popular Triple Crown races but developed into greatness later in their careers, horses like Forego, Fort Marcy, Tim Tam. Of course, Man o' War and Secretariat are there too.

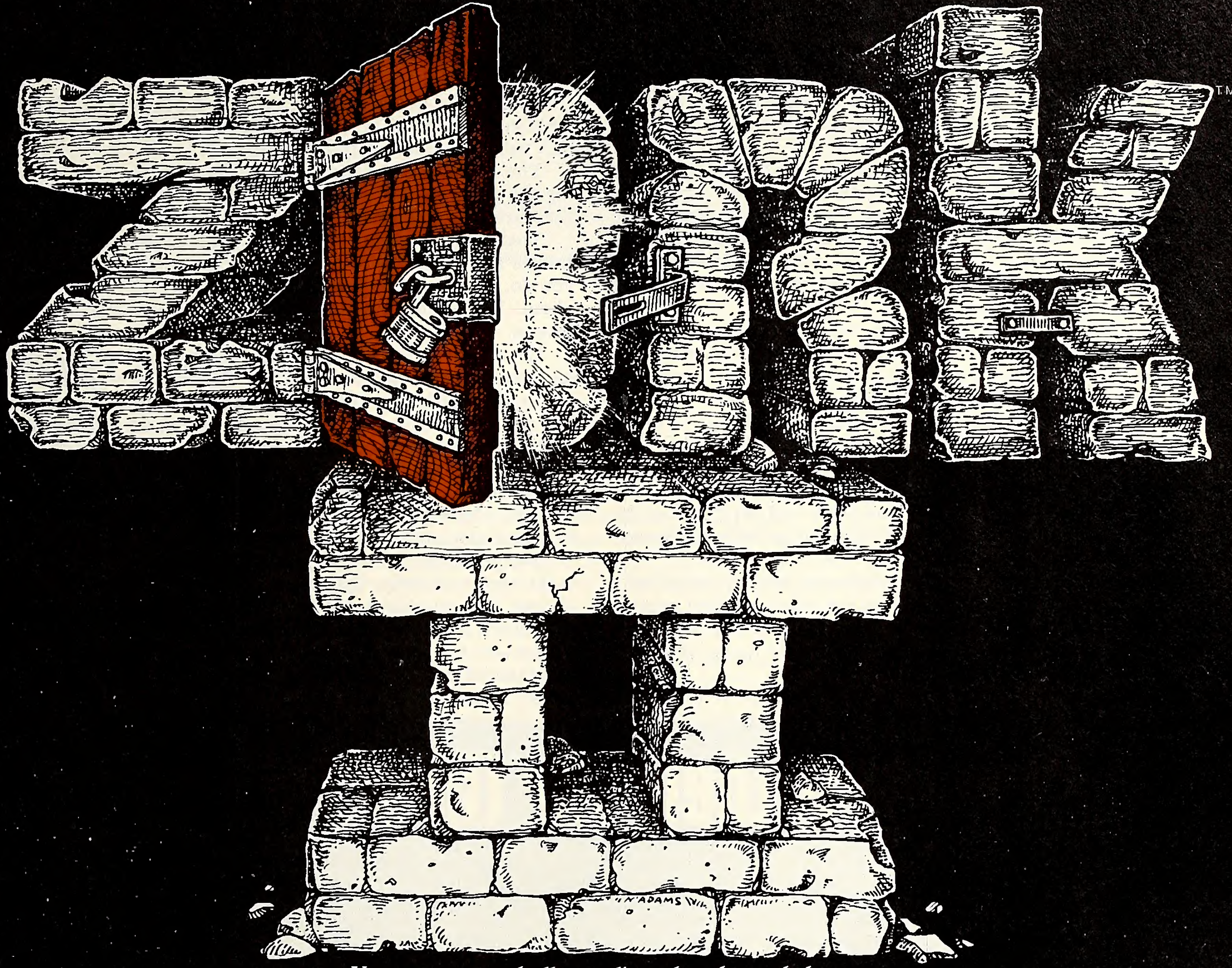
The program provides for as many as eight players and presents a ticket of ten races. Betting is excellently handled, although the odds are pretty odd sometimes, given the caliber of horses. Can you imagine Secretariat ever going off at twelve to one odds? But, then, maybe he would have, had he been racing against War Admiral, Citation, and Nashua. Thinking of these as normal races, the odds are very realistic and reflect about the right percentage of accuracy in outcome.

In addition to regular races—all presumed to be stakes or handicaps; purses are ignored—there are a daily double, a quinella, a perfecta (or exacta), and a trifecta to bet on. Payoffs are realistically accurate. Valentin clearly understands the totalizator board. You begin with a $200 stake and then anything goes. Not even the normal limits are placed on your betting. You can make one bet or several. You can bet fifty cents or all your money. No credit allowed, however.

Once the last bet is placed, the screen switches from the roster (which shows track condition, weights, and jockeys as well as horses and post positions) to the hi-res screen. This is again the best effort yet made for computer horseracing. Crowds of people are indicated on either side of the track. A tote board shows time of race and flashes win, place, and show numbers as the horses cross the line. There is even a steward with a flag that drops to start the race.

And here come the problems, the apparently insurmountable problems. The horses take turns moving, a spurt at a time. Reality is lost as a horse comes out of the gate after the field is halfway down the track and ends up taking the race. Even Silky Sullivan and Vigors started with the rest, albeit they fell back numerous lengths only to pass everyone in the

Just when you thought it was safe to go back underground again.



Your greatest challenge lies ahead—and downwards. Zork II: The Wizard of Frobozz, will be unlike any computer age adventure you've ever encountered. The underground world of Zork is designed to be lived and experienced in the most realistic sense. It features the largest vocabulary, the widest range of command options, the special capability to let you speak in complete sentences rather than two-word commands, and the most intriguing plot in the genre. And because Zork's mysteries are of the most challenging nature, it will take all your intellectual abilities to survive and emerge victorious from the Great Underground Empire.

Zork II is the extraordinary successor to Zork I, which hit #4 on the SOFTALK Top 30 in its first month on the market. Both run on 32K Apple II's with a 16-sector disk drive. And both are available now for Christmas.

The door to Zork beckons you. Look for it at your local computer store today.

INFOCOM

Zork is a trademark of Infocom, Inc.

Infocom, Inc., 6 Faneuil Hall Marketplace, Boston MA 02109

Apple II is a trademark of Apple Computer Corp.

stretch. Other horses run up to the finish line and stop, not even showing. There is method behind this madness, it appears; but let's get some animation that does the job right! Incidentally, the representation and individual animation are not a problem. The horses are tiny and well drawn.

Let's accept the erratic motion of the horses and assume the finishes indicate that they are running to form. Then there's a problem in play. On the graphic race screen, horses are identified only by the post numbers on the starting gate and at the finish line. This is just fine if you've memorized the race program—or, and this is a bit of a hassle, if you've written down the race ticket.

Perhaps you could overlook this omission—maybe there simply isn't room on the screen to get the names in—but when the payoffs come up on the tote board, the horses are still identified by numbers and no names. This means that, if you haven't written down the race and you've forgotten which is who, you're out of luck for finding out who won. Number 3 (or 6 or 1) is all you'll ever know.

Nevertheless, this remains the best race game, and, if you're willing to write down the horses in each race, you'll enjoy a pretty good day at the races with it. Anyone who wants to learn about betting and how the tote board works will find *Derby* a fine empirical teacher. DOS 3.2. $24.95.

☐ **Thief,** by Bob Flanagan, DataMost (Northridge, CA). With joystick you control a person traveling through a building of many rooms, each room peopled with several robots. Your job is to shoot the robots before they shoot you and get to one of the doorways and out before the berserk bouncing ball that relentlessly follows you can get you.

You can sometimes trick the robots and ball by positioning yourself by an exit in such a way that the bouncing ball gets all the robots before it gets to you, which counts as much toward your bonus as your shooting the robots would have, and then escape before the ball gets you.

When you reach an exit, the room scrolls off and a new one scrolls on. Rooms consist of wide outlines and partitions in varying colors. You and the robots are done in fine-lined animated hi-res. The big bouncing ball is just that with a grotesquely sickening grin that never changes—precisely as it should be. $29.95.

☐ **Copts and Robbers,** by Eric Knopp and Alan Merrell, Sirius Software (Sacramento, CA). *Copts and Robbers* places you in a haunted pyramid. Its resident ghost isn't scary at all; it just steals things.

This game defies categorization. It plays most like an arcade game, but it involves moving around a maze of which you can see only a small part at a time.

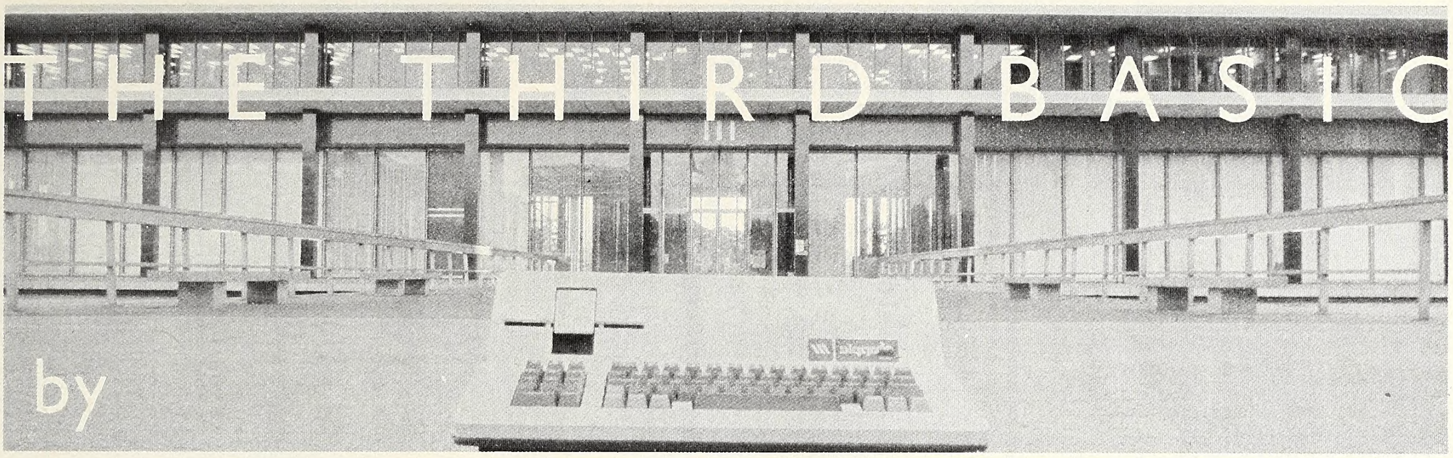
The objects and creatures—other than the one representing the player—are done in nice hi-res; but the maze itself is hires simulating lo-res.

The game goes like this. Your job is to find and return four jewels and a statue to a certain crypt within the pyramid. If you do, you win. The jewels are hidden in coffins that are lying about. You must find the key and open the coffins to find the jewels.

Level one is little more than practice in manipulating the keyboard. On levels two and three, the coffins may yield jewels or magic amulets or they may yield mummies that are out to get you. Also, the harmless but pesty ghost enjoys stealing things—including mummies, dead or alive—and putting them somewhere else.

There is a different maze for each level, but they never change. Once you're familiar with them, the whole game is avoiding the mummies and the ghost.

Game can be played at any of eight speeds. A great feature is the option to define the keys you wish to manipulate; you can choose the one-handed ijkm or return/right arrow/left arrow/slash diamonds or the two-handed arrows and a-z combination—or any other. Once again Sirius is to be applauded on this innovation. Game also has Sirius's now standard escape to pause clause. $34.95.



Taylor Pohlman

Exploring Business Basic, Part 3

Lots has happened to the Apple III since my last article, and I appreciate all your comments about the articles in this series. We'll have a chance to pick up on some of your suggestions next month, along with more news about the Apple III. For now we'll continue our exploration of the Business Basic file system as promised last time.

After reading this article and working with the examples, you should have a good knowledge of the differences between the *text* and *data* file types as well as more information about string handling functions and techniques.

We are going to stick with relatively simple indexing techniques for now, but next month we'll also cover some advanced indexing and file access methods to give you an idea of some of the ways that the popular data base programs retrieve data so rapidly.

Looking Back. The last article concluded with an example program that showed how the SOS file system could be used to store and rapidly find records in a file. We did that by using the random file access method that SOS and Business Basic have built in.

That technique allows file records to be numbered from 0 to 32767 and read directly without having to read all records from the beginning of the file.

The example at the end of last month's article also demonstrated that SOS uses a special storage and indexing method that wastes very little space in storing records on the disk, even if they have widely varying record numbers.

To go into further depth on this subject, and to compare the text file type we were working with last time to the more mysterious data file type, it's going to be necessary to create a more general version of last month's program. That example program allowed us to create a file that contained four pieces of information about a hypothetical parts distribution company. The data were the part number, description of the part, location in which the part was stored, and quantity on hand.

Unfortunately, the program was just designed to make some clever points about files, not to be really useful to parts companies. For example, we could only perform two functions, creating files and adding records. Most parts companies would want to look up parts, delete parts, get lists of parts, and so on. This month's version gets closer to that ideal, without denying you some of the fun of making your own changes. In addition, some of the functions that the program performs are generalized into subroutines so that we can make changes later without wholesale rewriting.

The New Parts Program. Well, now that you're breathless with excitement, here's the new version of the program:

```
5    HOME
7    PRINT
10   PRINT"Parts File Create and Modify Program"
20   PRINT:PRINT"Type:"
30   PRINT"      1 to Create a parts file":PRINT
```

```
40   PRINT"      2 to Use an existing parts file":PRINT
49   PRINT"      9 to Quit":PRINT
50   PRINT:INPUT"Your selection: ";a$
60   IF a$="" THEN 1000
70   a=ABS(VAL(a$))
80   ON a GOSUB 100,400
90   IF a=9 THEN 1000:ELSE 5
100  PRINT:INPUT"Name of new parts file: ";a$
110  IF a$="" THEN RETURN
120  CREATE a$, TEXT,64
130  PRINT"Parts file ";a$;" created."
140  RETURN
400  HOME
405  PRINT:INPUT"Name of existing parts file: ";a$
410  IF a$="" THEN RETURN
420  OPEN#1,a$
425  file$=a$
430  HOME
435  PRINT:PRINT"Modify Parts File"; CHR$(34);file$;CHR$(34):PRINT
437  PRINT"Type:"
440  PRINT"      1 to add to your parts file":PRINT
445  PRINT"      2 to delete a part from your parts file":PRINT
450  PRINT"      3 to find a part in your parts file":PRINT
460  PRINT"      9 to quit the modify mode":PRINT
465  PRINT:INPUT"Your selection: ";a$
467  a=ABS(VAL(a$))
470  IF a=9 OR a$="" THEN RETURN
475  ON a GOSUB 500,700,800
480  GOTO 430
500  HOME
505  PRINT:INPUT"Part number to add: ";a$
510  IF a$="" THEN RETURN
520  a=VAL(a$)
530  IFa<1 OR a>32767 OR INT(a)<>a THEN 500
535  rec=a
540  partnum$=a$
545  PRINT:INPUT"Description: ";a$
550  IF LEN(a$)>35 THEN a$=MID$(a$,1,35)
560  desc$=a$
570  PRINT:INPUT"Location: ";a$
580  IF LEN(a$)>15 THEN a$=MID$(a$,1,15)
590  location$=a$
600  PRINT:INPUT"Quantity on hand: ";a$
610  a=0:a=VAL(a$):IF INT(a)<>a OR a>99999 THEN 600
620  quantity$=a$
630  PRINT:PRINT"Record is: "; partnum$;"/"desc$;"/"location$;"/"
640  INPUT" OK? ";a$
650  a$=MID$(a$,1,1):IF a$<>"y" AND a$<>"Y" THEN 505
660  GOSUB 2000
665  IF errorcode=0 THEN PRINT:PRINT"Record added.":GOSUB 995:GOTO 500
670  PRINT:INVERSE:PRINT"Record not added, ERROR=";:NORMAL:PRINT error code:GOSUB 995:GOTO 505
700  HOME
705  PRINT:INPUT"Part number to Delete: ";a$
710  IF a$="" THEN RETURN
715  a=VAL(a$)
720  IF a<1 OR a>32767 THEN 700
725  rec=a
730  GOSUB 1800
735  If errorcode=1 THEN PRINT:PRINT CHR$(7);"No such part number":GOSUB 995:GOTO 700
740  PRINT"Delete ";partnum$;"/";desc$;"/" ;location$;"/";quantity$;"? ";
745  INPUT"";a$:a$=MID$(a$,1,1)
750  IF a$<>"y" AND a$<>"Y" THEN PRINT"Not deleted":GOSUB 995:GOTO 700
755  GOSUB 1900
760  PRINT:PRINT CHR$(7);CHR$(7);"Record deleted":GOSUB 995:GOTO 700
800  HOME:PRINT
805  INPUT"Part number to find: ";a$
810  IF a$="" THEN RETURN
815  a=VAL(a$)
820  IF a<1 OR a>32767 OR INT(a)<>a THEN 800
825  rec=a
830  GOSUB 1800
840  If errorcode=1 THEN PRINT:PRINT"No such part number":GOSUB 995:GOTO 800
850  PRINT:PRINT"Part number:      ";partnum$
855  PRINT:PRINT"Description:      ";desc$
860  PRINT:PRINT"Location:      ";location$
865  PRINT:PRINT"Quantity on hand:      ";quantity$
870  PRINT
890  PRINT:INPUT"Press RETURN to continue:", a$;GOTO 800
899  REM
900  REM delay subroutine
901  REM
```

```
995    FOR i=1 TO 1000:NEXT i:RETURN
996    REM
1000   PRINT:PRINT"End of parts file program."
1010   CLOSE
1020   END
1799   REM
1800   REM retrieve a record with record number =
       "rec"
1801   REM
1805   errorcode=1
1810   ON EOF#1 RETURN
1820   DEF FN scan(start)=INSTR(rec$,"/",start)-start
1830   INPUT#1,rec;rec$
1835   IF rec$="" THEN RETURN
1840   pointer=1:length= FN scan(pointer)
1850   partnum$=MID$(rec$,pointer,length)
1855   pointer=pointer+length+1:length= FN
       scan(pointer)
1857   Desc$=MID$(rec$,pointer,length)
1860   pointer=pointer+length+1:length= FN
       scan(pointer)
1870   Location$=MID$(rec$,pointer,length)
1875   pointer=pointer+length+1:length= FN
       scan(pointer)
1885   Quantity$=MID$(rec$,pointer,length)
1890   errorcode=0:RETURN
1899   REM
1900   REM delete a record with record number =
       "rec"
1901   REM
1905   PRINT#1,rec;""
1910   RETURN
1999   REM
2000   REM add a record with record number = "rec"
2001   REM
2005   errorcode=0
2010   rec$=partnum$+"/"+desc$+"/"+location$+
       "/"+quantity$+"/"
2015   ON ERR GOTO 2040
2020   PRINT#1,rec;rec$
2030   OFF ERR:RETURN
2040   errorcode= ERR:OFF ERR:RETURN
```

Well, nobody said that this series wouldn't get more interesting as we went along!

Let's take a quick look at the changes in this version of the program, as well as its major features. First, as to structure, the program looks something like this:

5-90	Initialization and first menu
100-140	Create a new parts file
400-480	Open an existing file and set up second menu
500-670	Add a record
700-760	Delete a record
800-890	Find and display a record
900-995	Subroutine to create a delay
1000-1020	Terminate the program and close files
1800-1890	Subroutine to find a record and load data values
1900-1910	Subroutine to delete a record physically
2000-2040	Subroutine to add a record with given data values

Note that for simplicity we have assumed a fixed file record structure. That is, we have *hard-coded* into the program the fact that the data items in each record are part number, description, location, and quantity on hand.

We have also coded into the program some restrictions as to the length of each item (lines 550, 560, and 610).

A real data-base program would have all this information stored in tables for more flexibility.

For example, there is no practical way, short of rewriting parts of the program, to add an extra data item to the records or change the meaning of the existing items.

Obviously, the more such generalizations we put into the program, the larger and more complex it will be.

Our purpose is to learn something about files first and then write the world's greatest data-base program.

To help understand the program and check out a few new features that make Business Basic really handy, let's look at the subroutines in the program.

First, examine the record retrieval routine at line 1800, which is used by the "Find" section and the "Delete" section. We will communicate any problems encountered in a subroutine by using the *errorcode* variable, with 0 indicating no error found.

The ON EOF statement in line 1810 will return with "errorcode" set to 1 in the event that the INPUT statement in line 1830 reads past the current end of file.

Line 1820 sets up a function definition that comes in pretty handy.

The function *scan* uses the Basic INSTR function to determine how many characters there are to the next occurrence of the "/" character.

Remember that we used the "/" character to delimit the fields within the string record we stored in the file. The INSTR function returns the character position of the string being searched for, starting with the position given by "start."

Subtracting the starting value from the position gives the total length of the field. More about INSTR can be found in the Business Basic manual. Give that section a look, because INSTR is one of the most useful functions you'll find. Some other Basics may use a different name for this function; POS is one example.

Line 1830 inputs the record according to the record number "rec." After checking for a "null" string (line 1835), lines 1840 through 1885 are responsible for breaking up the record into its separate fields. This is done by setting the variable *pointer* to the beginning of the field and then setting length to the number of characters in the field, using the scan function defined previously. The MID$ function is then used to make the assignment to the appropriate variable.

Study this section carefully to be sure you see how this works. One technique to understand routines like this is to make a diagram of the data and work through the statements while playing computer.

Now that the individual fields are assigned to the proper variable, they can be used in the calling routines (at lines 730 and 830) to display the values as desired. Later on we are going to change the structure of this file considerably, and it will be handy to be able to handle that by changing the routine at 1800 rather than making changes throughout the program.

The delete routine at line 1900 is really simple, just consisting of printing a

null record over a previously existing record. As we change the file structure, this may become much more interesting.

The add routine is also simple, consisting (at line 2010) of packing the various field values together using the String Concatenation Operator (the world's longest way to refer to "+"). There is one thing of interest, however. Note that the ON ERR statement is used to trap any errors which may occur in writing to the file. We again use errorcode to communicate that an error has occurred and are careful to turn error trapping off before returning to the main routine.

It would have been possible, and even desirable, to use the ON ERR statement to check for all errors in the program, but the routine required to make the program that bullet-proof would have made this program unnecessarily long. It's probably a good subject for a future article.

Well, now that we've been through the major features of the program, we suggest you enter the program and start fooling around. As we mentioned last time, this program was never meant to be the ultimate in user friendliness or elegance of coding style. As you add records, find, and delete them, try to imagine ways you could improve the way the program works or asks for information.

Business Basic DATA files. While it's certainly true that most files contain data, Business Basic uses the term *DATA files* in a special way. You may remember that a TEXT file consists of strings of characters with the carriage return character as the terminator between strings.

If you print a numeric variable into a text file, it will automatically be converted to a string value, just as is done when printing numbers to the screen. This sounds pretty nice, but it can cause some real problems and inconvenience. For example, you know that an integer variable (which ends with the % character) occupies two bytes of storage in memory. However, representing the value in string format can take up to six bytes (−32000, for example). Add a RETURN character to delimit it and you have up to seven bytes to store an integer in a file.

Furthermore, it's impossible to tell beforehand just how much space a given set of numeric variables will take without checking each one beforehand. This can cause design problems for programmers. As you can imagine, these problems are even more acute for the long integer data type, which can be up to nineteen digits long but only requires 8 bytes of internal storage.

There's another problem with using text files that shows up only when you are using real numbers. Reals are represented in Business Basic as 32-bit floating-point quantities requiring four bytes of internal storage. Normally, they are displayed with six digits of precision, and the format itself may vary greatly—especially if the magnitude of the number is very large or very small. In those circumstances, Basic will display the number in scientific notation. This means that the output format of a real can vary from something simple like 3.45 to something like −1.36723E−06.

Interestingly enough, it's not so much the space that this notation takes up that causes the trouble but that the printed representation of a real may not correspond exactly to the value stored in memory. If a number's representation is not exact or requires more decimal places than can be displayed, the number is rounded before printing.

By contrast, this does not occur with integers. Since rounding occurs during printing, and text files are storage of the printed format, values of real numbers may be different in the text file than they were in memory. A short example will illustrate:

```
10  OPEN#1,"numberfile"
20  INPUT"type two numbers: ";x,y
30  z=x*y
40  PRINT#1,1;z
50  INPUT#1,1;z1
60  IF z=z1 THEN PRINT"they compare":GOTO 20
70  PRINT"they don't compare: ";z,z1
80  GOTO 20
```

Note that by printing the value to the file with the random access method in line 40, we are able to read it back directly in line 50. This lets us check to see if any value change has occurred as a result of the file operation. Try this with values like 500 and 4.25. Everything should go normally.

Now try a value like 3.033 and .031. Still okay. Now try 3.031 and .031. The result should print out appearing exactly the same, yet the comparison in line 60 fails. If you wish, you can insert a statement at line 75 to print out the difference. It will be small but obviously significant. For the real reason the product of this number pair fails to work, we commend you to your local math professor or textbook on numerical analysis.

Suffice it to say that certain real numbers cannot be stored exactly as binary numbers, nor can certain binary numbers be displayed exactly in a finite number of digits. As soon as these situations occur, the quantities stored in the text file will not exactly match what was calculated in memory. Play around with this program further. There's almost an infinite number of combinations that will also fail the test but appear to be equal.

You've just seen two reasons for the need, from time to time, to store numbers in a file in the exact form they have in memory. Can you think of a circumstance where you might want to do that with a string?

Among others, if you have a string that contains (or could contain) a return character, the text file input statement will terminate wherever the return oc-

curs, thereby losing the rest of the characters in the string.

The key is that with DATA file format, you can store any numeric or string quantity without worrying about what might happen to the information. In addition, Business Basic adds an identifier to the front of each item of data you store in a data file, to indicate what kind of data it is. This is called the data Type, and allows you to intermix integers, reals, and strings in any order and still read them back correctly.

The information about the type of a particular data item is retrieved, astoundingly enough, by the TYPE function. This allows a simple program to read the contents of any data file, without having any information about it beforehand. Much more information about data files can be found in your manual, and I suggest you spend some time reviewing it.

In the meantime, let's look at what using data files will do to the parts program I listed at the beginning of the article.

First, we'll need to change the file type specification on the CREATE statement at line 120. The new line will look like this:

```
120  CREATE a$, DATA,64
```

Since the program was fairly modular, with the file access done in subroutines, the other changes are minimal as well. The idea is to store each item we used before (part number, description, and so on) as a separate data item in the file. Since the part number is always a four-digit number, we can use an integer to store that data. Description and location are string quantities, and quantity on hand will fit nicely into a real value, since it's a maximum of five digits (line 610 checks for that).

The first subroutine to change is the one at line 2000, which writes a record. The new statements look like this:

```
2010  partnum%= VAL(partnum$):quantity=
      VAL(quantity$)
2020  WRITE#1,rec;partnum%,desc$,location$,
      quantity
```

There, that was easy.

Note that WRITE was substituted for PRINT since this is a data file, and instead of packing all the strings together as we did in the old line 2010, we simply converted the string values to the appropriate numeric ones.

If we had designed the program to use data files from the beginning, we probably would have handled that in the program's data entry section.

Next come the changes to the subroutine that reads a record back. Now things are very simple. We can replace all the lines between 1820 and 1885 with these:

```
1815  READ#1,rec:IF TYP(1) = 5 THEN RETURN
1820  READ#1,rec;partnum%,desc$,lacatian$,
      quantity
1825  IF partnum%<0 THEN RETURN
1830  partnum$=STR$(partnum%)
1840  quantity$=STR$(quantity)
```

That's it! Since all the items are stored separately, there's no need to go through the process of splitting them out of the string record.

We must confess, however, that we really wanted to discuss the INSTR function, and that prevous technique seemed the most logical way to show its features. Oh, well, it's always more fun to find an easier way!

Two more things are of interest here. Note that we have checked in line 1815 for TYPE 5, which indicates end of file. This takes care of checking for empty records. In line 1825, we introduce a new concept.

Previously, when we wanted to delete a record, we simply printed a null string over the existing information. There are times when it's useful simply to flag that a record is deleted, not actually wipe the information out. This allows deleted information to be retrieved in the event of mistakes.

Periodically another routine can be used to go through the file and physically delete the flagged records. Here, and below in the actual delete routine, we make the part number negative to indicate that it's no longer an active record.

The "delete" routine now will look like this:

```
1905  partnum%=-partnum%
1907  WRITE#1,rec;rec;partnum%,desc$,location$,
      quantity
```

That will write the record out with a negative part number, which will flag it as logically deleted.

Well, that should just about do it. In addition to giving you several things to try out before next time, the above changes illustrate an important programming fact of life. (You've been waiting for this article to get juicy, right?) This fact is that the more modular your program design, the more painless it is to make inevitable changes. I know that isn't your favorite fact of life, but there's nothing worse than to stare at several thousand lines of Basic, knowing that it has to be completely rewritten.

Next time we'll cover some new but related topics that will require completely rewriting this month's program. (Just a joke!) Actually, we'll talk about different ways to store and retrieve records on disk that give more flexibility than the simple record number scheme used so far. That should complete the effort to make you a file expert. In addition, Business Basic has an incredible output formatting capability, and now that you have learned the techniques for storing data, it should be fun to go through some tips on how to make your printouts look like professional reports.

Until then, have fun practicing the facts of life—programming facts, of course.

☐ **Raising an Issue from the Past.** On April 14, 1912, the British luxury liner *Titanic*, proclaimed to be an unsinkable ship, hit an iceberg at twenty-three knots in the north Atlantic. Within two and a half hours, the sixty-six thousand-ton vessel sank to the bottom of the sea, taking with it more than fifteen hundred victims and leaving in its wake a slew of mysteries surrounding the accident.

Since then, several adventurers have attempted, without success, to find the sunken wreck. Probably no one has come closer than the crew of the R.V. *Gyre*, which sailed with computers aboard.

Led by millionaire Texas oilman Jack Grimm, a team of oceanographers and computer experts embarked on a two-week expedition last July, sixty-nine years after the fateful maiden voyage of the S.S. *Titanic*. Their goal was to find the hull of the ill-fated ship in a sector approximately the size of the state of Rhode Island—a needle-in-a-haystack search requiring some of the most sophisticated technological equipment available.

Computers navigated the *Gyre*, plotted maps, and stored location constants about the search area, approximately two hundred fifty miles off the coast of Newfoundland. A sonar vehicle, towed by the *Gyre*, combed the water—thousands and thousands of feet of it.

Digital image processing was used to make a picture of the ocean floor from the sound waves. When an unusually large mass appeared on the computer screen, video cameras were dangled over the side of the ship. All resulting information was deposited in the computer.

Grimm has since announced that his underwater cameras photographed what he believes to be the propeller of the *Titanic*. If it is, says Grimm, "the hull of the ship isn't far away." However, others point out that several ships have sunk in those waters.

Even if Grimm's discovery isn't the *Titanic*, the adventurer is not discounting the importance of his voyage into history. "There are so many mysteries surrounding the sinking of the *Titanic*. Data from 1912 is pretty fuzzy."

Grimm plans to make a documentary film about his expedition and is considering a return to the area in a submarine next summer to get a close look at what he hopes is the *Titanic*. And, if he goes, computers will again be included in the search. There are, after all, a lot of mysteries still left to solve, countless hypotheses to be proven or disproven, and maybe some riches to find. "There are some irresistible qualities in such deep waters," adds Grimm.

☐ **For Your Eyes Only.** A blink of an eye may now be enough to prevent the wrong person from getting secret information stored in a computer or sent through electronic mail. The Eyedentifyer, whose inventors are saying is a more reliable personal recognition device than fingerprints, may very well revolutionize computer security. Blood vessel patterns in the eyes of each person are unique, says inventor/researcher Robert Hill.

When the Eyedentifyer system is hooked to a computer, it will automatically screen those who will eventually receive the communication. The system resembles a movie camera and requires the user simply to blink into it once to put his unique eye signature on file. For example, instructions contained in electronic mail may stipulate that this message is for certain eyes only. When the communication arrives, the person on the receiving end blinks into the camera. If the eye signature is on file, the message is decoded; if it's the wrong pair of eyes trying to sneak a peak, the Eyedentifyer will keep the message to itself.

Hill's father, an opthalmologist, contributed to the research. Eyedentify, Inc., says brain-eye interaction—the key to the invention's success—has several future uses in the medical field. Detecting glaucoma and eye damage from diabetes in the early stages may eventually be possible with the Eyedentifyer.

In the meantime, the invention is being kept under wraps while being tested and improved. And, according to Eyedentify, "Due to the technology's proprietary nature, the company does not wish to elaborate further at this time."

Despite the veil of secrecy, Eyedentify, Inc. will probably have to elaborate on one burning issue sooner or later: If an electronic mail user has a hangover, will that prevent him from getting his correspondence?

☐ **Still No Apple.** When we last left Gottfried R. von Kronenberger in his cell block at Folsom Prison (see August 1981 *Softalk*), he was writing a stack of letters, trying to solicit donations for the inmates' computer science program. Since then, von Kronenberger has sent out over three hundred letters to the heads of the largest United States business corporations, who have proven responsive to the needs at Folsom; some of the finest computer hardware available has been sentenced to a life term at Folsom. Several new printers, dumb terminals, and one entire business system have been donated to the program by leading corporations, including Diablo, Vector, Centronics, and Xerox.

While computers are still in short supply, interest in the computer science program among the inmates at Folsom is not lacking: about twenty-two are enrolled in computer courses, with more than one hundred on a waiting list. The steadily growing program is also receiving so much support from prison administrators that the waiting list could soon be a thing of the past.

Computer science has already been allocated a larger teaching area; it formerly shared a limited space in principal Gigg Powers' office, but it's now headquartered in a downstairs classroom of its own.

Education superintendent Robert Miller recently announced that, starting in the spring quarter, Folsom Prison will offer associated arts degrees in both data processing and computer science. The latter program will emphasize computer repair.

Inmates are also eligible to enroll in a Bachelor of Arts program at Folsom, and Miller says he's considering making a semester course in computer science one of the requirements toward that degree. "I think all of our students who graduate should have computer literacy in an age where it's going to become a necessity," he says.

But von Kronenberger, who started the entire program, probably won't be around Folsom to watch the first inmates receive their degrees in computer science. He goes before the parole board this month and stands a chance of being released before Christmas.

So how does a man who's already served almost ten years survive the last couple of months? "I don't even have time to think about it," admits von Kronenberger. "I'm too busy here—writing all those letters."

☐ **Game-Playing for a Fuller Life.** If laughter is the best medicine, then Dr. Bill Lynch is writing the right prescription by bringing video games into the hospital. Lynch, who is program director for the Brain Injury Rehabilitation Unit at the Veteran's Administration Medical Center in Palo Alto, California, believes that popular computer games might be doing more for his patients than hours of traditional therapy.

Lynch is testing this hypothesis on twelve patients who have physical impairments as a result of traumatic accidents, strokes, or heart failure. Treatment for such patients usually includes speech, physical, and occupational therapy; further therapy for cognitive functions such as concentration, hand-eye coordination, and reaction time is usually accomplished through video games.

The good doctor believes so much in the power of video games that he has written a small resource guide about the types of games available and the cognitive abilities they help to strengthen. Adventure games, for instance, are useful for solving problems related to visual searching, strategy, and anticipation of consequences.

All the sports games, including computer basketball, bowling, and football, require a knowledge and application of game rules, strategy, hand-eye coordination, and quick reaction time. The various computerized card games such as *blackjack* and *poker* require players to polish up their math skills in counting card values and determining wager amounts.

Games simulating driving greatly improve motor function, and joystick-maneuvered games can train patients to wield the joystick on an electric wheelchair. They can also help determine whether a patient is physically coordinated enough to drive a real car—not on a race track, just in normal traffic.

Lynch and his colleagues have been gathering data on normal subjects and on adults with assorted brain lesions. Scores are being recorded, and a standard data base, much like the regular personality or neuropsychological data already in hospital records, is being compiled. These profiles in turn help provide ongoing feedback to patients throughout therapy.

Although a number of hospitals and medical centers around the country are using video games in a therapeutic context, there is not a lot of published research in the area.

Lynch's research in California has not yet produced sufficient data for Lynch to feel absolutely secure in announcing a medical breakthrough. But he does point to the work of a New York researcher, Dr. Renee Okoye. Working with twenty-five children with learning disabilities, Dr. Okoye had seven of the children play games at thirty-minute intervals, once a week, for a period of three months. Her research showed dramatic increases in the children's abilities since they were first evaluated in the Southern California Sensory Integration Tests, which measure motor accuracy, bilateral motor coordination, and spatial visualization.

Lynch says the games provide enjoyable competition for patients, as well as another way to pass time during a long hospital stay. Best yet, his patients feel no threat from watching the television screen. Video games "are also colorful, they make noise, and the patients don't see it as hard work the traditional way."

In spite of budget cuts, Lynch is trying to scrape together some funds to buy more games. He's also looking into the possibility of research grants to fund the purchase of more computers and software to enlarge the project so more patients can play. Lynch and his enthusiastic staff also figure they might start developing their own software to fit individual needs of their patients. Any way the program is supported, it seems to have an electric future at the Palo Alto V.A.M.C.

"There will always be a need for the human clinician," says Lynch, "but the need for more efficient and economical treatment will require a change . . . to a combination of group, individual, and automated techniques."

☐ Computers are taking the guesswork out of auto maintenance. The West German auto manufacturer BMW has equipped some of its cars with a service interval indicator—a small computer that complains when the car needs attention. By keeping track of the distance and time driven, number of starts, and engine age, the computer alerts the driver if a checkup or oil change is needed with a series of flash-on dashboard lights. After the car has seen its local mechanic, the computer can be reset, and the driver can start putting more wear and tear on the engine. ■

It's a Jungle Out There!

THE BOOK
WILL CLEAR THE WAY THROUGH THE SOFTWARE TANGLE

Don't settle for manufacturers' promotional material.

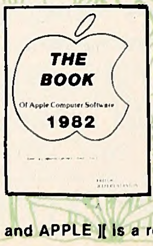
THE BOOK is a complete **critical analysis** of most Apple software available. Experts **review, rate** and **evaluate** the programs on 11 separate points. It covers Games, Education, Business and Utilities as well as hardware.

Don't buy software until you read this book. With the overwhelming array of programs available you can't afford not to consult ***THE BOOK.***

Swing by your local computer store or order direct.

TheBookCompany

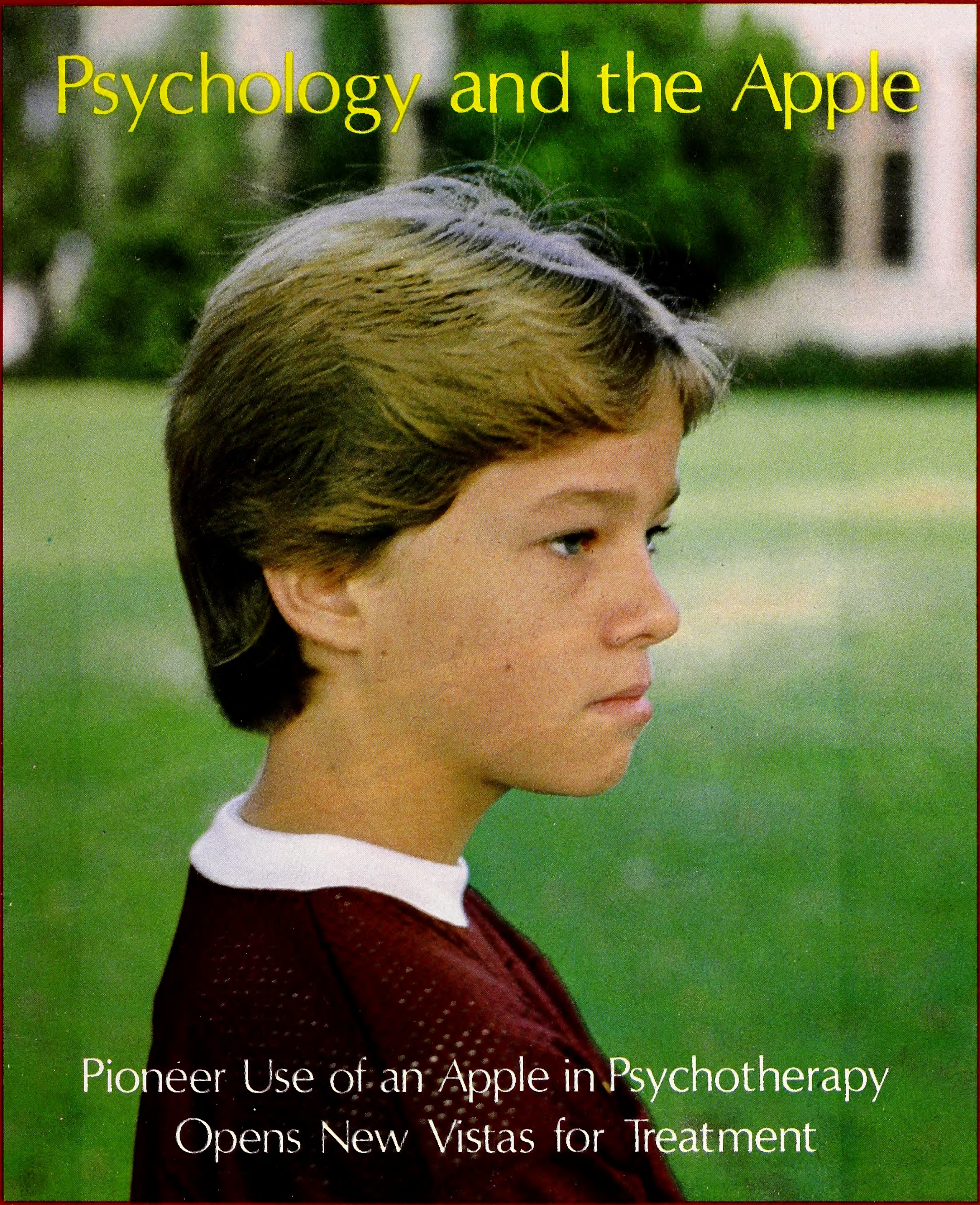
16720 Hawthorne Blvd.
Lawndale, CA 90260
(213) 371-4012



NAME ____________________
ADDRESS ____________________
CITY __________ STATE __________ ZIP __________
CARD NUMBER __________ EXP __________
MasterCard ☐ Visa ☐

APPLE and APPLE][is a registered TRADEMARK of APPLE COMPUPTER, INC.

$19.95



Psychology and the Apple

Pioneer Use of an Apple in Psychotherapy Opens New Vistas for Treatment

BY KATE DICKSON & ZEV WANDERER

When Dennis's parents brought their fourteen-year-old son to Dr. Zev Wanderer for psychotherapy, they were distraught. Their bright, towheaded son couldn't seem to keep out of trouble. He had committed a number of antisocial acts, including the theft of a neighbor's motorcycle. He was prone to temper tantrums at home, had been caught cheating on exams at school, and was unresponsive to his parents' efforts to help him.

Clinicians at one of the finest child counseling centers in Los

Zev Wanderer, Ph.D., a Diplomate in clinical psychology, is in private practice in Malibu, California, as a marriage, family, and child counselor and is founder of the Center for Behavior Therapy in Beverly Hills. Wanderer is coauthor of the bestselling book Letting Go: A Personal Action Program To Overcome a Broken Heart *and of* Making Love Work: New Techniques in the Art of Staying Together.

Kate Dickson is a freelance writer in Ventura, California.

Don't Let Your Apple Miss the Boat

On June 5, 1982, a luxurious cruise ship will depart from Vancouver, Canada, with several Apples on board. For seven days, these privileged Apples will be learning such arcane arts as assembly language from Roger Wagner and graphics from Ken Williams, as well as compiling Applesoft programs into assembly code with Dennis Goodrow.

Besides coming back more intelligent, the Apples will be treated to the usual shipboard conveniences by a professional staff dedicated to providing everything a microprocessor might desire. The Apples will enjoy some of the most dramatic scenery north of Silicon Gulch, stopping in Ketchikan, Juneau, and Skagway, some of the last frontier communities. They'll cruise past Glacier Bay—giving the Apples a look at a true hi-res graphic.

Amazingly, in keeping with Softalk's policies, the Apples will be taking the tour free. Apples craving the companionship of their owners may inquire about the cost of human participation by writing to:

Softalk Cruise
Attention: Donna Siebert
11021 Magnolia Boulevard
North Hollywood, CA 91601

This tour is exclusively arranged for Softalk readers by Valencia Plaza Travel Agency of Newhall, California. Roger Wagner, Ken Williams, and Dennis Goodrow are committed to giving seminars on this cruise unless extraordinary business contingencies arise.

Angeles had diagnosed Dennis as needing intensive psychotherapy; four sessions each week for at least two years was their recommendation. What a job! thought Wanderer. How can this unhappy, unresponsive child be helped more quickly?

At first, Wanderer had no more success in working with Dennis than his parents had had. Dennis refused to take part in therapy; instead, he eyed the clock in Dr. Wanderer's comfortably appointed consulting room and demanded to know "if I can leave now."

Wrong Exit. The treatment of troubled adolescents has some unique aspects. Children in general, and delinquent young people in particular, tend to have limited attention spans. They are more likely to manifest their pent-up emotions through play than through the introspection and talking out typical of adults. Delinquent youth, especially, tend to have high levels of tension and to act out to relieve their psychological discomfort.

Such impulsive acts as aggression against parents, siblings, and peers are classic examples of an unhappy youth's acting-out behavior. It is not surprising that delinquent children have low frustration tolerances and limited coping skills. Behavior that appears to be simple cruelty or selfishness is often a cry for help, using the only ways the child has learned to get people's attention. Often, parents, teachers, and others who don't understand the pain that underlies and motivates these objectionable displays of behavior write such a child off as a bad kid.

The child therapist applies no such label. The so-called bad kid is simply a kid with problems—problems that must be dealt with to unearth the good kid. The therapist's task is to help bring forth the normal, developing personality the obnoxious facade has so successfully obscured.

Play therapy was the strategy Wanderer wanted to try with Dennis. In play therapy, therapist and client participate together in recreational activities. For example, the pair might play a game of checkers together and then walk over to the local fast food stand for hamburgers. During the time they share, the therapist observes the client, rewarding the positive aspects of behavior on the spot. The therapist also makes an effort to modify the negative aspects of the client's behavior.

But, although play therapy held promise for solving one set of problems, it presented another. Because Wanderer suffers from a severe visual handicap, excursions and games requiring sharp eyesight weren't possible. Whatever progress was to be made toward helping Dennis would have to take place in Wanderer's hilltop Malibu home. But how could play therapy be used there?

Making Goodness Pay. Meanwhile, the immediate problems resulting from Dennis's delinquent acts were serious and pressing. Temper tantrums and verbal aggression at home as well as noncompliance with rules at school were not the only issues. Recently, Dennis had defaced a neighbor's property with canine feces, smeared in the form of a swastika. He had been unjustly accused by the neighbor of vandalism and had responded in the only way he knew how—with a violent, antagonistic act.

To a behavior therapist, managing an individual's behavior involves assessing the rewards (or payoffs) of a particular behavior and then modifying the reward structure as necessary. Deficits, such as Dennis's aggressive, dishonest actions, usually call for an expanded, prosocial reward structure. For example, if Dennis were to be stopped from stealing money from his mother's purse, he would have to be given more powerful positive reinforcement for not stealing. He would have to learn to *want* to behave differently.

Dr. Wanderer believed that increased self-esteem would be a powerful reinforcer for Dennis, and that if Dennis were given opportunities for self-improvement, his self-esteem would increase. Accordingly, the initial therapy was to consist of four

sessions each week, one of which would serve as a parents' feedback session.

In theory, the therapist's idea was excellent. Practically speaking, however, it was a disaster. Dennis was constantly saying "I want this! I hate that! No, I won't do it!" Dennis was not bad, stupid, or mean, the therapist realized; he just didn't know how to negotiate, how to achieve his goals in a smooth, effective, harmonious fashion.

Enter an Apple. Well, neither do many adults! thought Wanderer, resolving to find a way to help Dennis. The key was to motivate Dennis to want to develop such negotiation skills. "Underneath his restiveness and anger is a strong, intelligent personality. Dennis can get what he wants, and fairly. But he must learn how to negotiate for it."

Then the dawn—a breakthrough—or what Dr. Wanderer describes as a "serendipitous find"—the Apple II computer.

Dennis often complained of being deprived (which, in truth, he was not). Comparing himself to his classmates at private school, he declared that "almost everybody except me has his own computer."

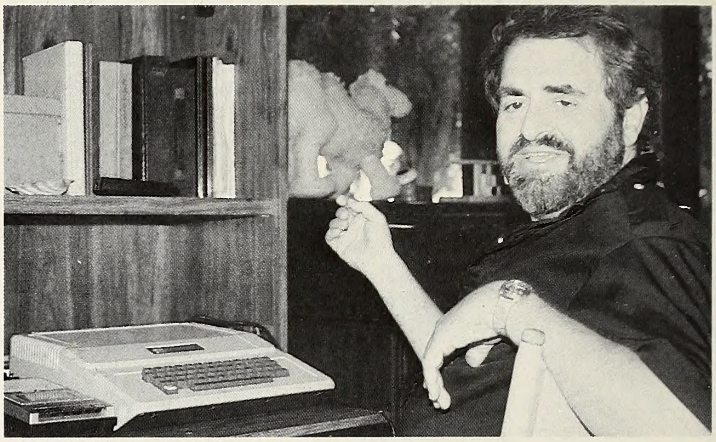
Wanderer, an Apple owner, recognized the perfect opening when he saw it. Perhaps the computer could function as the medium for the conversations he and Dennis couldn't have together in the therapy room.

The therapist struck a deal with his young client: in exchange for five minutes of true participation in therapy during each session, Dennis would receive computer time. While five minutes may not seem significant, it's a long time to a restless, troubled adolescent.

The pact worked surprisingly well. Once Dennis knew that the therapist would keep his end of the bargain, clock-watching ceased. And, while Dennis enjoyed his well-earned computer time, Wanderer counseled his parents, offering strategies, such as contingency contracting, that they could try out at home.

How could a modern gadget, a home computer, effect such a change in a boy's behavior? Follow us . . . .

Bomber, by Bob Bishop (Softape), was the first game Dennis played on the Apple. In *Bomber*, tanks progress along the horizon at a preset rate of speed while bomber planes fly overhead at a different fixed speed. The object of the game is to destroy tanks. But since the tanks keep moving, and the bombs go forward and down, a player doesn't score hits by simply releasing bombs when the plane is directly over its tar-



Dr. Zev Wonderer

get. Becoming a skilled bomber player involves learning to think ahead, taking into account both the speed of the moving tanks and the speed, direction, and trajectory of the bomber planes. Originally, the therapist saw *Bomber* as merely a game Dennis could play, but it soon became evident that *Bomber* would be an important component of the play therapy.

One of Wanderer's main concerns was Dennis's extreme impulsiveness. When Dennis wanted something, he insisted upon immediate gratification. Since no one has all desires met on demand, it's easy to imagine the problems that resulted from Dennis's tantrums and impulsive, antisocial behavior. Family members and friends loved Dennis, but they felt alienated from him and did not know how to deal with his behavior.

Game Strategy Wins in Real Life. When Dennis first played *Bomber*, he was unable to control his impulses and tapped the keyboard continually. Soon, however, he realized that, in order to win, he had to exercise control over the bombs. To increase the efficacy of Dennis's learning, Wanderer asked Dennis to pretend that each point he lost through impulsive tapping cost him five cents. Thus, Dennis was encouraged to plan his bombings and began to experience the success that results from patient, strategic moves.

Wanderer, who specializes in the treatment of addictions of all sorts, says that Dennis became, in a sense, addicted to the Apple. Finally, an activity that was truly rewarding for Dennis had been found, and positive behavioral change was evident in therapy within one week. In addition, Dennis's parents reported that their son's ability to delay gratification had begun to be apparent in other aspects of daily life. Now, for instance, when Dennis asked his mother for a special snack food, he could agree to wait until an appointed time for it. Quite a change from his earlier tantrums.

At this point, Dennis's parents became cotherapists with Wanderer, who directed them to reinforce their son's prosocial behavior with both tangible and social rewards. While money and special privileges are powerful rewards for children, social rewards such as praise and encouragement are even more powerful. Right now, Dennis was receiving both types of rewards; later on, social rewards would dominate.

The behavioral child therapist uses play therapy for three purposes: diagnosis—to observe behavior; strategy—to gain attention by participating; and therapy—to help the child learn skills such as cooperation and sharing through play. The therapist notices, for example, how the child reacts to losing a game and tries to convey to the child, "It's okay to lose some-

times. That doesn't take anything away from your worth as a person. Everybody loses. I lost the last game we played together, but I feel fine, because we had a good time playing."

It was learning to lose with grace that led to an increase in Dennis's self-esteem. Said Dennis, "You won. I did a crummy job; you did better." "Yes," Dr. Wanderer replied, but you played well, and I really enjoyed your participation. Thanks for the fun we're having together playing these games."

Conspiring to Win. Each new game Dennis played on the Apple helped him to develop more prosocial attitudes and behavior. To play *Star Wars* (Apple Computer) as a two-person game, players must work together using the two game paddles to center the enemy ships in the sights of a laser gun. Once they have centered their target appropriately, they are ready to fire missiles at their adversaries. Scores mount through effective joint effort and are lost through noncooperation with one's partner. Dennis quickly discovered that problems are not solved by blaming others or having temper tantrums. Instead, he learned cooperation.

Initially, Dennis was unwilling to talk about his own strategic plans, and Wanderer's goofs made him angry. Finally, Wanderer said, "You know, I can't read your mind, Dennis. If you communicate, if you tell me what moves you'd like to make, then we can work together. Maybe I'll make fewer mistakes then. But I'll still make some goofs, I know, so it will make us both feel better if you don't get so angry about them."

Thus, the skills of negotiation, of give and take, began to manifest themselves in Dennis's behavioral repertoire. Not surprisingly, his parents reported that communication at home was better as well.

The next game Dennis played was *Dragon's Maze* (Creative Software). In this game, a single player, the blue dot, is pitted against the dragon, a red dot. The dragon builds a maze, and the player's goal is to maneuver his way through the maze, steer clear of the dragon, and escape.

When Dennis first started playing *Dragon's Maze*, he was especially concerned with how his scores compared to Dr. Wanderer's. As he continued playing the game, Dennis began to see that what was important was his own progress and improvement. Instead of being preoccupied with how his score compared with someone else's, Dennis began to enjoy competing with himself. He began to see that life is not a war with others, nor is antagonism a necessary component of interaction.

Thus, *Dragon's Maze* functioned as a tool for self-improvement. Dennis was highly motivated to keep his cool and play wisely, and he could see that, as Dr. Wanderer pointed out, "Points are gained in this game, as in life, by calm appraisal."

An incidental bonus from the *Star Wars* play was evident in the next game Dennis played, *Tank War*. In *Tank War*, players get to choose the size and shape of their own tanks. Then, each player attempts to maneuver his tank into good firing position, while staying out of range of the other player's shots. Points are scored when a hit is made and lost when shells are fired and miss their mark.

Dr. Wanderer's errors in *Star Wars* were usually caused by his limited vision, a condition Dennis knew about. The boy's growing esteem for his therapist revealed itself in a new character trait, compassion. Realizing that the therapist would enjoy *Tank War* play more if a larger tank were his target, Dennis suggested that he (Dennis) control a larger tank. That way, Dr. Wanderer would have a larger target to shoot at and would present a smaller target since he would control a smaller tank. Strategically, Dennis put himself at a disadvantage in *Tank War* play; psychologically, though, his increased self-esteem was a real winner.

Metamorphosis. Thus an angry, aggressive, delinquent youth changed dramatically in less than six weeks. Wanderer reports that since the incorporation of the Apple into therapy, Dennis's delinquent activities have virtually ceased.

Dennis's characterological qualities are of particular interest. On the few occasions when he has been dishonest recently, he has confessed to his parents. Dennis's parents have also established a negotiation system for both behavioral and academic performance. Points are earned by honest and productive acts and lost by dishonest and antagonistic ones. Furthermore, Dennis's ability to delay gratification and increase his frustration tolerance by participating in therapy before having computer time is consistently rewarded with praise by both parents and therapist. In addition, Dr. Wanderer includes the cognitive aspect of the therapy by pointing out what is going on via comments like, "Do you notice how, when you wait, when you are patient, you get more joy, and it costs you less? So it is in life, Dennis."

Although he now actually requests time with Wanderer, Dennis will soon graduate from psychotherapy. His newly developed negotiation skills have helped him achieve the ability to handle life's daily frustrations. The increased levels of communication at home make Dennis especially happy. Even his sixteen-year-old sister has responded positively to her once-impossible brother.

Instead of intensive psychotherapy, four times a week for a minimum of two years, Dennis's therapy and dramatic behavior changes have taken place over the course of four months. Now, fifteen-minute sessions once every two weeks are all that's necessary.

One might well ask how, since Dennis was originally referred to long-term psychotherapy, such a change could occur so early. A psychoanalytic therapist might object, for instance, to the fact that Wanderer and Dennis spent so little time discussing the origins of Dennis's problems. Why was he so angry and alienated in the first place? Others might object to the fact that so much of the time for therapy, which should be serious business, was spent playing computer games.

There's No Time Like the Present. There are some simple rejoinders to these concerns. First, while behavior therapists may consider it interesting to delve into the origins of self-defeating actions, they don't accept such exploration as essential to the therapy. They see the problems the client is having in the present as more pressing and urgent. And the emphasis in the therapy is on finding ways to help solve these problems. How, the behavior therapist asks, can negative behaviors be discarded for self-enhancing ones in the most rapid manner? How can tangible reinforcers, such as money and gifts, give way to social ones, such as praise and encouragement?

For the behavior therapist, the answer is simple. Most people are affected more profoundly by the emotional responses of the people they care for than they are by concrete rewards. Intrinsic rewards, such as feelings of self-worth and self-esteem, are much more powerful than extrinsic ones.

Dennis's behaviors and attitudes are becoming increasingly prosocial. He enjoys earning his new 3.2 grade point average at the private school he attends, a new closeness with his sister and parents, and his relationship with Wanderer—not because of the tangible payoffs, which are gradually being phased out—but because it makes him happy to relate cooperatively to people. Both he and the people who are important to him clearly derive the benefits.

Recently, Wanderer asked Dennis how he felt about himself. Dennis responded, "I like myself better." This increased self-esteem is, in Wanderer's estimation, the most important factor effecting the positive change. And once self-esteem begins to take root, it has a delightfully healthy chance to grow.

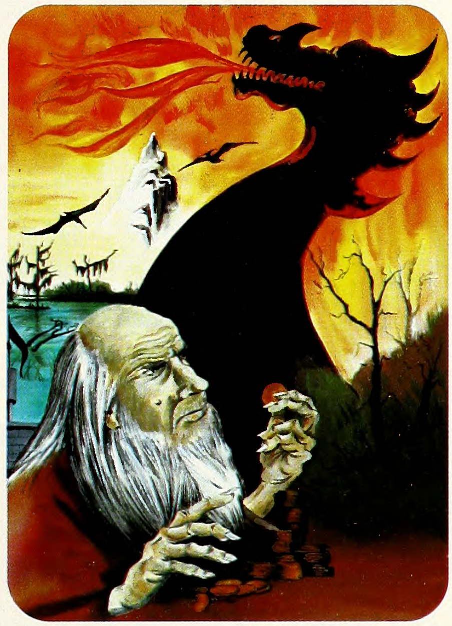
Sharing responsibility for Dennis's swift rehabilitation are his parents, who cared enough to bring their son into therapy and to act as cotherapists, and who continue to provide him with opportunities to experience success; the creators of *Bomber, Star Wars, Dragon's Maze,* and *Tank War*, sources of both learning and fun; and an amazing new therapeutic tool, the Apple II.

But, of course, Dennis is the most responsible of all.

TARTURIAN

THE TARTURIAN requires 48K RAM, APPLESOFT ROM, and disk. As you explore the 160 rooms (each done in HI-RES) gathering weapons and treasure that will prepare you for the final battle against the TARTURIAN, you will encounter deadly KROLLS, battle the MINOTAUR, try and get by COUNT SNOOTTWEEKER, decipher the YUMMY YAKKY'S secret, make friends with the TULIESWEEP, avoid GHOULS, explore the PILLAR tombs, discover secret passages and more, 5 interlocking programs.

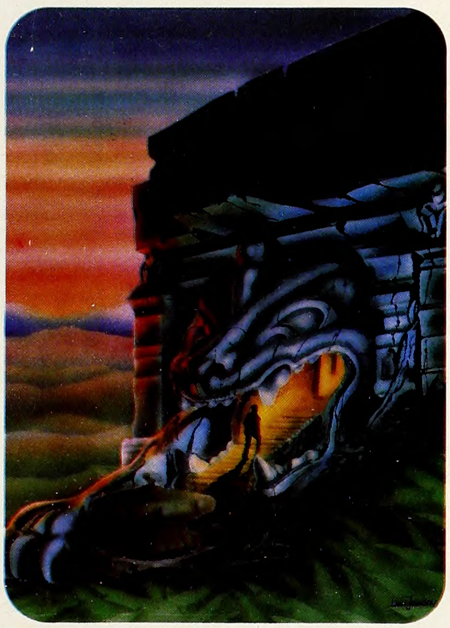
TARTURIAN on disk **$24.95**



GOBLINS

Escape to GOBLIN COUNTRY, an adventure to challenge your skills and cunning where you will need skill and sorcery to survive DRAGONS, OGRES and GOBLINS. Myths and magic, treasure and danger. Fantastic High Resolution Graphics and animation help you on your journey. Requires 48K APPLESOFT ROM and disk.

GOBLINS on disk **$27.50**



CREATURE VENTURE

You have just inherited your Uncle Stashbuck's mansion but first you must rid it of the horrible creatures that have taken it over and find your uncle's buried treasure. Directing the computer with two word commands such as 'Go North', 'Get Key', 'Look Room', 'Punchout Boogeyman' etc., you will need to explore deep into the mansion to finally find the Stashbuck fortune. There are tons of High Resolution graphics plus some clever animation just for fun.
Requires 48K RAM, APPLESOFT ROM and disk.
All High Resolution characters generated with Higher Graphics II by Robert Clardy.

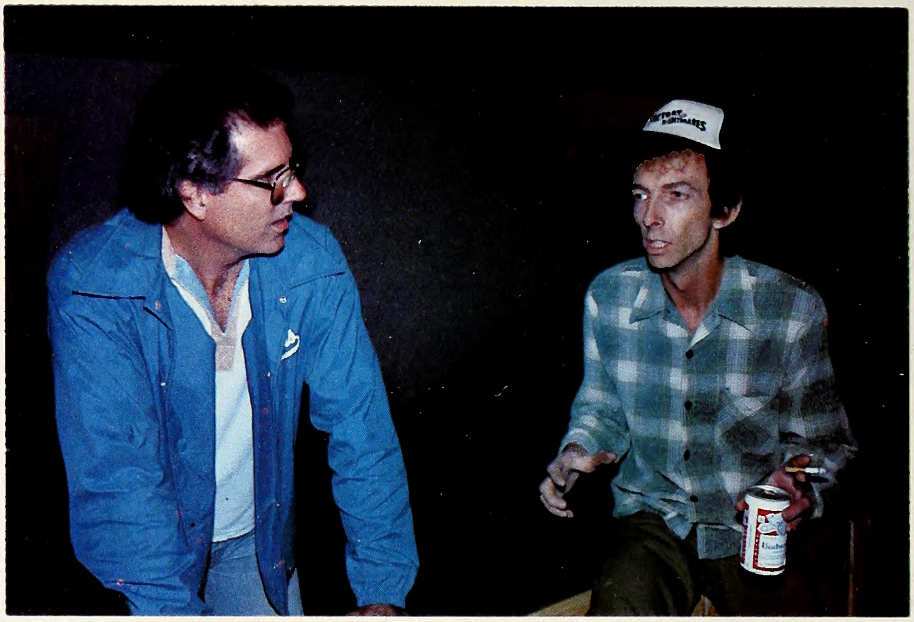
CREATURE VENTURE on disk **$24.95**

SEE YOUR LOCAL DEALER OR SEND CHECKS TO
HIGHLANDS COMPUTER SERVICES
14422 S.E. 132nd
Renton, Washington 98056
(206) 228-6691

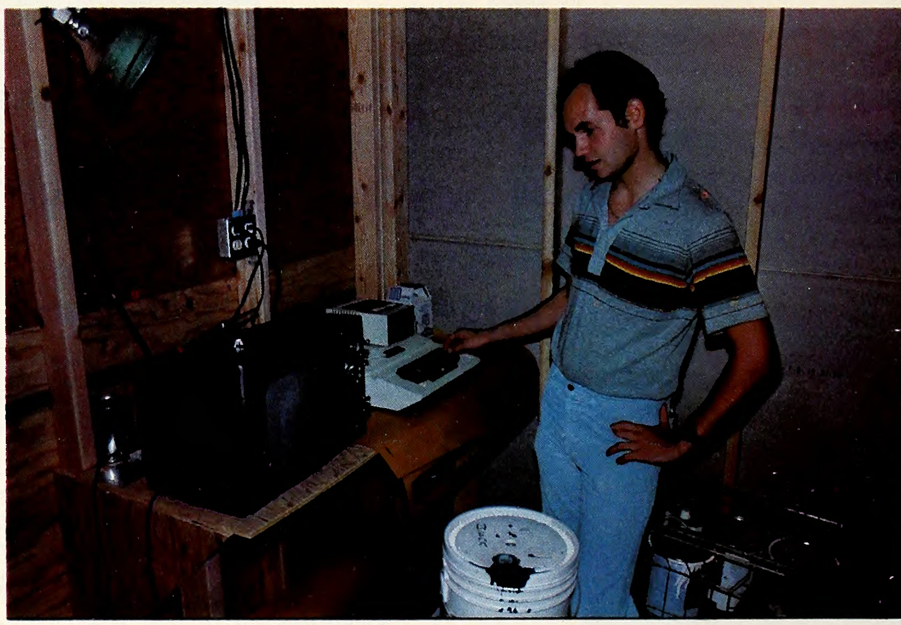
VISA

master charge

Washington residents add 5.4% sales tax. Applesoft and Apple are registered trademarks of Apple Computers, Inc.



Haven't you seen these men somewhere before? Maybe in a nightmare? Charlie Butler (left) and Steve May, members of the Northridge, California, Jaycees, undergo a dramatic change each Halloween.



Eric Waller, UCLA college student, Northridge Jaycee, and computer expert, programmed his Apple to provide all the scares.

A *Haunted* Apple

BY MELISSA MILICH

In its prime, it was a beautiful house, a happy place where the owners could relax and reflect on how lucky they were to live in such nice surroundings, where visitors were invited to rest after a journey and enjoy all the comforts. But one night an earthquake disturbed the magnetic balance of the area, and the house shuddered as if it knew what to expect. One by one, the inhabitants disappeared, and people no longer came to visit.

A long time passed. Trees in the front yard leaned over and died, and brambles and weeds choked the gardens. The townspeople refused to remember the cursed house, and it settled well into obscurity.

One starless October night, a rat chanced upon the structure and entered, hoping to find a good meal. It crept through the darkness and sniffed the ground, searching for a crumb. Suddenly its whiskers wrinkled. A horrible stench permeated the room, for the house was by no means empty. A shrouded figure seized the rat and eyed it hungrily. The rat squirmed and stared, fascinated. It had never seen a human like this before: One eye dangling out of a decayed socket, a crooked mouth full of crooked, corn-colored teeth. *Was* this a human? The rat didn't care; it turned and bit the hairy hand and managed to free itself—just barely. It scurried away in the darkness, and the shrouded figure screamed after the retreating rat. It was fresh blood that kept his teeth yellow and his body odor intolerable. He would find it somehow.

When you first meet Charlie Butler, he looks vaguely familiar. You're sure you've seen him somewhere before . . . but, no—he has attractive blue eyes, even, white teeth, and he smells okay, too. Butler is an active member of the Northridge, California, chapter of the Jaycees, a fraternal organization dedicated to the strongest American ideals. There are no monsters in Butler's closet—harmless ghouls and goblins, maybe.

Since this time last year, Butler has been chairman of the Factory of Nightmares, an annual fund-raising event sponsored by the Northridge Jaycees. During a two-week period each Halloween for the last seven years, the Factory has produced nightmares, and for the last three years, an Apple computer with a ghoulish bent has gotten into the act.

The factory is more than just a haunted house. "No drippy little spooks or things that go bump in the night for us," Butler brags.

Some of the techniques are so sophisticated, the scenes so real, and the characters so gruesome that those who visit the factory have little doubt while they're there that what they're seeing is the real thing.

"We've had people ignore the exit doors and tear right through the walls. This is thick wood," and Butler pounds it for emphasis, "but it hasn't stopped some really scared people."

Just how does this happen? Only the Apple knows all the secrets.

Nightmares are divided into several sections and rooms, including the swamp-graveyard, the checkerboard room, the butcher shop, the torture chamber, the Halloween room, and the rat room. The factory is huge, about sixty-five hundred square feet in all, and approximately fifteen hundred visitors run through it a night.

Hundreds of Jaycees and volunteers are needed for crowd control, security, and tickets. There's also first aid (some who don't get out in time faint), and cleanup (for those who should have fainted). Some of the luckier Jaycees are chosen to play the spooks in the house: Dracula, Frankenstein, the Werewolf, and assorted ghouls.

As for the Apple—it runs the rest of the show. In a room all by itself, the Apple controls light switches, leaving the audience in total darkness at appropriate times; plays tapes of eerie music (the haunted house top ten); flies a head down a hallway; and cues the actors to start playing their scenes.

The last function is particularly important at the Factory of Nightmares. There is an element of theatre in this haunted house. The Jaycees take great pleasure in taking a leading role in many of the nightmares. The Butcher Shop, for instance, is a good place to avoid. There, a couple of maniacal workers go merrily to their task, chopping up bodies and throwing them into a giant-sized meat grinder.

Is this real, the audience wonders? Just then, the head butcher leaps over the railing, grabs a juicy-looking human from the audience, and carries his live hunk of flesh back to his work table. "We use a lot of audience participation here, but the audience doesn't know it till they get here."

When it's time for the actors to take their parts, a disk drive hidden out of audience sight turns on its light. This subtle cue tells the butchers, restless spirits, and dead bodies who rise out of coffins that they have only fifteen seconds to do their job till the computer leaves them—and the audience—in total darkness.

The program was developed by Eric Waller, an employee of Rainbow Computing in Northridge, and its timing guarantees that the crowd will keep moving—either through the rooms or straight through the wall.

The Factory of Nightmares accomplishes its purpose. The ghostly effects seem to frighten the visitors sufficiently enough, in fact, that they keep coming back year after year for more. The factory raises several thousand dollars annually, all of which is donated to needy children, Boy Scout troops, and other worthy causes.

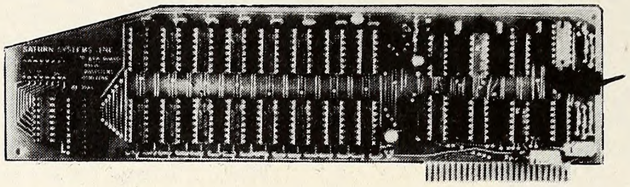
If the Jaycees didn't insist that most of the victims in the torture rooms and butcher shops be women, it would be a very worthy—and just as frightening—evening.

The factory of nightmares will return same time next year for those who enjoy the threat. "You'll see the reflections of your nightmares," promises Butler.

Do you smell a rat?

SATURN SYSTEMS 32K RAM BOARD FOR APPLE

32K of buffered read/write memory on a plug-in card



Compatible with: Apple II®, Apple II+®, Microsoft's Z80 Softcard®, DOS 3.2, DOS 3.3, INTEGER Basic®, Applesoft®, PASCAL, FORTRAN, LISA®, Personal Software's VISICALC®

Software included:
1. Relocation of DOS into SATURN 32K board (recovers appproximately 10K of main board RAM).
2. Utility package for saving and loading Applesoft® and INTEGER® programs and data on the 32K RAM board; overlaying, chaining.
3. PSEUDO-DISK: Modifies DOS 3.3 to allow use of SATURN 32K RAM board(s) like another disk drive.

COMPREHENSIVE DOCUMENTATION • 1 YEAR WARRANTY

ALL FOR ONLY $239.00

NEW!

MEMORY EXPANSION SYSTEM FOR VISICALC®

Now you can expand the memory available to Personal Software's 16 sector VISICALC® using the SATURN 32K RAM BOARD!

With VC-EXPAND™ and one or more SATURN 32K RAM BOARDS the memory available to VISICALC® is increased from 18K to:

50K with 1 SATURN 32K BOARD
82K with 2 SATURN 32K BOARDS

VC-EXPAND supplied on 16 sector disk

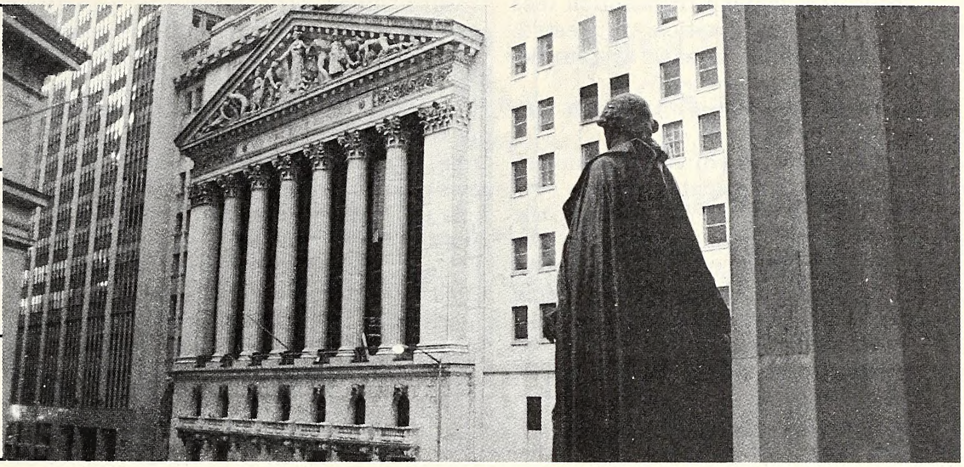
ONLY $100.00

SATURN SYSTEMS

SATURN SYSTEMS, INC.
P.O. BOX 8050 • ANN ARBOR, MI 48107 • (313) 665-6416

Buttonwood Apples

BY KEN LANDIS



The first step in making an investment decision is gathering information. This can be an informal process—such as cocktail party conversation—or detailed, painstaking research into the particular investment under consideration. Either way, the gathered data will tell whether further investment consideration is warranted.

Even after the investment is made, the investor still has a need for timely, precise information. As investors, we all have different informational needs based on our own investment criteria and techniques, and our data requirements determine which services and software are most suitable.

The monetary value of the investment also determines the software and services required. A small investment won't justify the expense of frequent use of any service, but an individual who follows the markets closely and trades frequently on short-term price movements may be able to justify the cost of frequent use of one or more services. The rule of thumb for making this decision is a cost benefit analysis; if you are making a satisfactory rate of return that would not be possible without the service, then it's worth the price.

The three major services that provide financial information electronically to the microcomputer user are the Source, CompuServe, and Dow Jones. To understand the issues and make comparisons, the following glossary is provided.

Glossary

Baud rate: the rate at which the computer transfers information, measured in bits per second.

Bit: binary digit; the smallest unit of information the computer recognizes, represented by the absence or presence of an electrical impulse.

Data base: computerized files of information

Keyword: a word or phrase that the computer will search for.

Response time: the time it takes a system to execute a command after it has received it.

Telenet and Tymnet: nationwide computer communication networks.

Information Services

The Source, owned and operated by *Readers Digest*, is a large multidata base system. All stock, commodity, bond, metals, and money market information on the Source is provided by United Press International, which supplies financial information to a large segment of the news media. There are 168 UPI Unistox reports available to Source users on a daily basis. The system also provides business headlines from the *New York Times*. Both the UPI and *New York Times* data bases can be used to extract current information on world, national, state, and local events as well as business.

Information on Demand (IOD) is also available on the Source. It's a research service that—given your specifications—will prepare a research report on any subject you choose. Minimum cost is two hours at $40.50/hour. The Source does not warranty information received in the reports.

Raylux Financial Services Reports are prepared weekly by economists, investment advisers, and other financial experts. There are two sections to the report, business outlook and financial commentary. These interesting, informative reports are recommended for background and general overview purposes.

Foldex is a program through which the Source allows you to access the UPI Unistox data base and retrieve the latest quotes on your portfolio automatically. This is accomplished by creating a list of the securities, commodities, or metals in your portfolio and directing the Source to retrieve this information. The *New York Times* data base and UPI wire service are available.

The Source user's manual is easy to read and informative but does not cover every command that may be used on the system, or the system's nuances. If you have problems, the customer support staff is quite helpful, and there is rarely a wait of more than four minutes before you can communicate with them.

Cost of the system is reasonable, although there is a $100 one-time registration fee, which may deter some potential customers.

System response times have been fairly rapid since the Source recently underwent a major expansion program and installed much faster equipment.

Although the Source offers no historical stock price data, it can be a valuable information resource for the investor.

To learn more about the Source's financial information, enter the command DATA BIZDEX after the Source command prompt. When using this command, have a text storage program such as *Data Capture* or *ASCII Express* loaded in your Apple. It's a rather long file; you'll want to save it—and possibly print it out—for later reference.

The CompuServe Information Service, owned by H&R Block, is another large multisubject data base. One subsection of the system, called Microquote, contains financial information. Included are trading statistics and descriptive information on over 32,000 stocks, bonds, and options. Historical trading data is available on most securities from January 1, 1974, to the present. Historical dividend records are also available from January 1, 1968, to the present. Data on options is available only for the currently traded options, not those that have expired.

CompuServe charges additional fees for connection to the Microquote data base. There is a base charge of $1 each time the investor connects to the service. This is in addition to the basic connect

fees outlined at the beginning of this section. There are varying transaction fees based on the information requested. These fees are listed in each system description that follows.

Data and Programs Available on Microquote.

Cusip is an acronym for Committee on Uniform Security Identification Procedures. This program will identify the ticker symbol or Cusip number for any equity, debt, or option issue. Transaction charges for this program are twenty-five cents for each twenty-five listed issues.

Price is a program list that gives daily, weekly, or monthly trading statistics or dividends for any given security. All prices and dividends listed are adjusted for stock splits and/or stock distributions. Price quotation information is the date, volume (in 100s), high/ask, low/bid, and close. Dividend distribution data is also available and includes date, rate, type, ex-dividend date, record date, and payment date. The fees for this program are $.05 per price set for daily data, $.10 per price set for weekly data, and $.15 per set for monthly data. Dividend data cost $.15 per period for each security.

Examin is a program consisting of comprehensive statistics of the issue, whether it be a stock, bond, or an option. The information given includes such items as the Standard and Poor rating, shares outstanding, beta, latest bid, price and dividend history, Moody's rating, bond rate, yield, maturity date, open interest, exercise price, and underlying issue price. The cost of using *Examin* is $1.25 per issue.

The *Stats* program gives the high, low, close, high close, low close, volume, mean, and standard deviation of the high, low, close and volume of an issue over a given period of time. The transaction cost of the *Stats* program is $1.25 per issue for a time period of up to 261 trading days (one calendar year) or $2.50 per issue for time periods greater than 261 trading days.

Quotes provides price information for a group of securities for a single date. Prices are adjusted for stock splits and dividends; the data given for each security is ticker symbol, high, low, close, volume, and *Cusip* number. The system allows you to create a file of ticker symbols or *Cusip* numbers to use as input to *Quotes*. This program is similar to the *Foldex* option on the Source. Cost to use *Quotes* is $.05 per security priced.

Modata allows investors to create files of historical stock prices or current quotes and transfer them to their own systems by having files typed out via modem connection or by using the *Filtrn* (file transfer) program. Program use is explained when on-line with CompuServe or by the customer support staff.

Additional Financial Information on CompuServe:

Raylux Financial Services Reports—the same reports carried on the Source.

Commodity News Service (CNS), a Kansas City-based firm that supplies the CompuServe network with current commodity prices and news stories.

Archers Commodities, a Chicago-based company that supplies strategic outlooks on a variety of commodities.

Standard and Poor's Information File—descriptive financial information on many publicly held companies.

Value Line Data Base II—contains more than 400 different descriptive items on companies in the industrial, utility, retail, banking, and insurance industries.

The following newspapers are on-line: the *San Francisco Examiner* and the *San Francisco Chronicle*, the *New York Times*, the *Washington Post*, and the *Los Angeles Times*. Financial news from the Associated Press is also available.

Unfortunately, the 800 numbers to CompuServe's customer support center are extremely busy, and delays in reaching this center have run up to thirty minutes. The staff is courteous and knowledgeable, and response times are good, but log-on time can be very slow, with two to four minutes not unusual in the evenings. The user's manual is short but adequately covers system operation. The CompuServe package is marketed through Radio Shack stores. It's inexpensive, has a wide variety of financial information, and—if it's the investor's first exposure to time sharing—a valuable experience and a good way to test the waters.

The Dow Jones data base is a service of Dow Jones and Company, publishers of the *Wall Street Journal* and *Barron's*. The Dow Jones news/retrieval data base is a cornucopia of financial information. Services available are:

Current quotes for stocks, bonds, mutual funds, U.S. Treasury issues, and options (minimum fifteen-minute delay).

Historical quotes for the last trading month, monthly summaries from 1979, quarterly summaries through 1978.

Money market services weekly economic survey compiled from surveys of economists and money market dealers at some of the nation's top financial firms. The survey is recommended for general background and overview readings.

Disclosure On-Line, with company profiles, SEC 10-K extracts available on over 6,000 companies. Company profiles contain company headquarters addresses, SIC, codes, and descriptions of companies' businesses.

News, containing news articles and composites from the *Wall Street Journal*, *Barron's* and the Dow Jones News Service. You may scan the headlines and select stories that reach as far back as ninety days or ninety seconds. The news is classified into eighty categories, more than fifty industries, and 6,000 companies.

Media general financial services is a data base updated weekly, consisting of detailed corporate financial information. The service includes fifty-two statistical indicators on each of 3,200 companies and 180 industries.

Free text search covers more than 125,000 articles as far back as June, 1979. Searches may be made by keywords.

Transcripts on-line from *Wall $treet Week* includes transcripts from the last four programs. It's divided into four categories: Mr. Rukeyser's commentary, panel discussion, viewers' questions and answers, and the guest interview.

The Dow Jones news retrieval service is a data base that supplies not only basic trading data but also sophisticated tools for the serious investor or analyst. The availability of the *Wall Street Journal* and *Barron's* on-line is a valuable asset to any serious student of the financial markets.

The *Wall $treet Week* transcripts are an interesting novelty item. They may, of course, be obtained through the mail, but

Dow Jones

Supplier: Dow Jones and Company (Princeton, NJ)
Operating hours: 6 a.m. to 3 a.m. EST
Customer service hours: 9 a.m. to 11 p.m. EST, weekdays; 9 a.m. to 5 p.m. EST, Saturday
Accessibility: through Telenet, 9 a.m. to 5 p.m. weekdays, and Tymnet, all hours of operation
Prime time: 6 a.m. to 6 p.m. weekdays

Rates per minute:	*Prime time*	*Nonprime time*
Continental USA		
News	$1.00	$ .20
Quotes	$ .75	$ .15
Historical quotes	$ .75	$ .15
Wall Street Week transcripts	$ .50	$ .25
Disclosure On-Line	$ .83	$ .83
Free Text search*	$1.33	$1.33
Media General	$1.00	$1.00
Money Market Services	$1.00	$1.00

Storage Charges: not applicable
Minimum monthly charge: none
Registration fee: none

**Available 8:30 a.m. to 8 p.m., Monday; 8 a.m. to 8 p.m., Tuesday through Friday*

The Source

Supplier: Source Telecomputing Corporation (McLean, VA)
Operating hours: 7 a.m. to 4 a.m. EST, daily
Customer support hours: 7 a.m. to 4 a.m. EST
Accessibility: through Telenet or Tymnet or direct within a 50-mile radius of McLean, VA

Hourly rates:	*300 baud*	*1200 baud*
Continental USA		
Weekdays		
7 a.m. to 6 p.m.	$15.00	$25.00
6 p.m. to 12 p.m.	$ 4.75	$ 9.00
12 p.m. to 7 a.m.	$ 2.75	$ 6.00
Weekends and holidays		
7 a.m. to 12 p.m.	4.75	$ 9.00
12 p.m. to 7 p.m.	$ 2.75	$ 6.00
Hawaii, Alaska, and Canada		
Weekdays		
7 a.m. to 6 p.m.	$15.00	$25.00
6 p.m. to 12 p.m.	$15.00	$14.75 (Hawaii) $19.75 (Alaska) $12.25 (Canada)
12 p.m. to 7 a.m.	$15.00	$13.25 (Hawaii) $18.25 (Alaska) $10.75 (Canada)

International charges vary according to connecting network. Source hourly charges for international users are the same for those in the forty-eight contiguous states.

Storage charges: .0165 cents per day per block (2048 characters)
Registration fee: $100
Minimum monthly fee: $10

if time is of the essence, the Dow Jones news/retrieval service is a fast, convenient way of obtaining these reports.

The documentation/information that Dow Jones sends to new subscribers is complete, concise, and easy to use. There is an instruction sheet that takes you step by step through the log-on procedure, using a D.C. Hayes Micromodem.

The customer support staff is extremely helpful and ready to answer any questions you may have about the service. During our tests, we had some trouble with the commands on the system, but customer service was able to get us on-line quickly. System response time is good, even during heavy traffic periods.

Dow Jones is by far the most expensive of the services to use. For that reason, the Dow Jones news/retrieval service is a tool for serious investors who wish to have at their fingertips the wealth of information and expertise that is Dow Jones and Company.

If you don't have a modem or wish to gather the information manually, two excellent sources are the Market Laboratory found in *Barron's* (Dow Jones and Company) or Standard and Poor's *Stock Quotes*. Information may also be gleaned from your local newspaper's financial section. Always be sure you're recording the closing information for the securities you're tracking. If you use interim information, you won't have the correct data on which to make an informed decision.

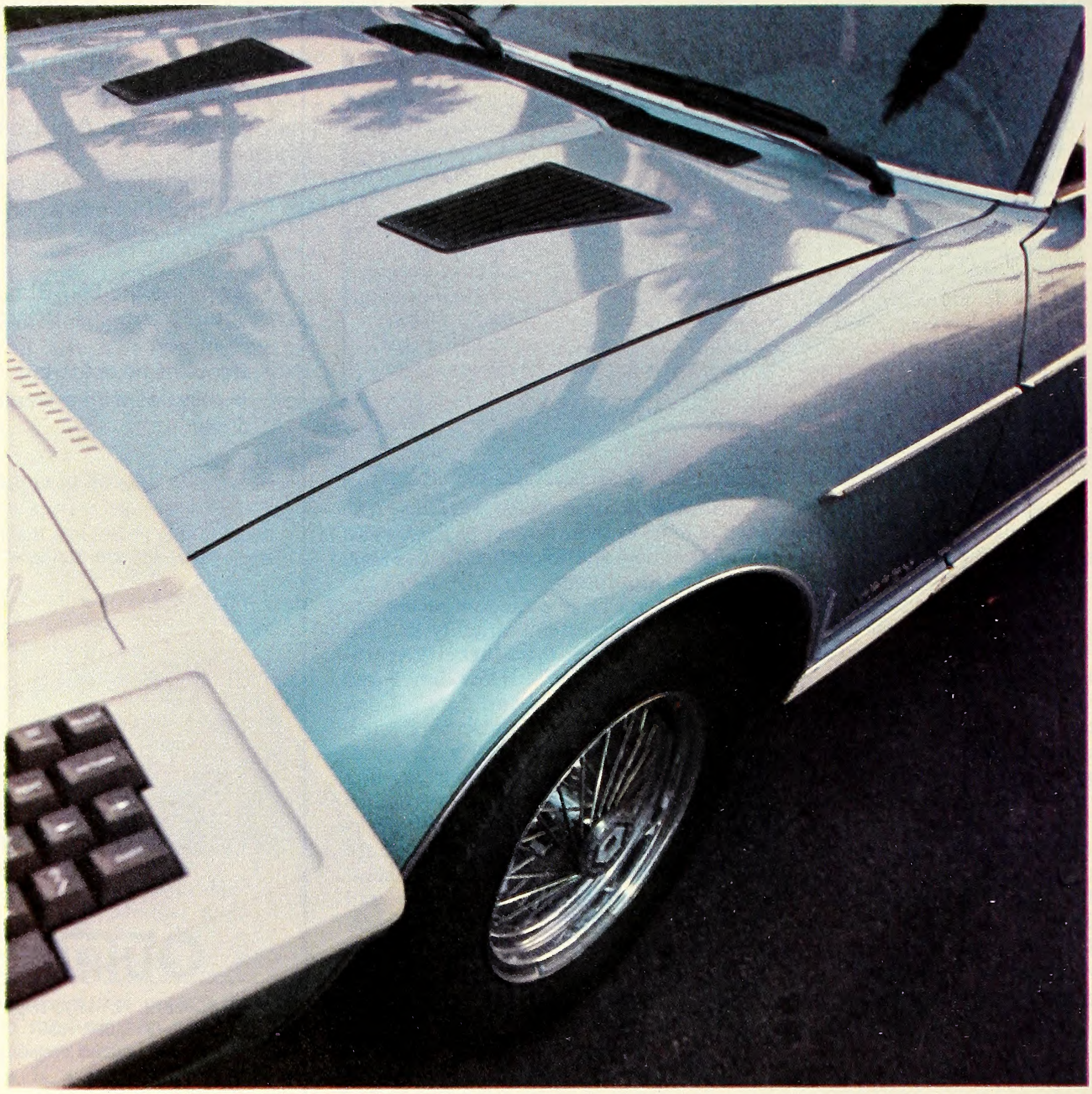
The next article in the series will cover the software packages that directly use the financial information collected. ■

CompuServe

Supplier: CompuServe (Columbus, OH)
Operating hours: 24 hours a day, 7 days week, unless maintenance is scheduled
Customer support hours: 9 a.m. to 12 p.m. EST, Monday through Friday
Accessibility: through the CompuServe network and Tymnet (through Tymnet add an hourly surcharge of $10 in prime time and $2 in nonprime time.)

Hourly rates:	*300 baud*	*1200 baud*
Continental USA		
Weekdays		
5 a.m. to 6 p.m.	$22.50	*not available*
6 p.m. to 5 a.m.	$ 5.00	22.50
Weekends and holidays		
24 hours a day	$ 5.00	$22.50

Storage charges: 128K free; increments of 128K $8.75 per week
Registration charge: subscriptions available through Radio Shack stores (one hour free with subscription). Two packages are available: dumb terminal package, for use with any modem with terminal firmware or software, $19.95; Videotex software package (terminal software), $29.95
Minimum monthly charge: prime service, $45; no minimum for nonprime service



D A V I D

TELECOMMUTING

HUNTER

As I was walking that ribbon of highway,
I saw above me that endless skyway,
I saw below me that golden valley,
This land was made for you and me.
Woody Guthrie

James Martin, in his book *The Wired Society* (Prentice Hall, 1978), drew the analogy that telecommunication networks will be the highways of the future. Through the use of computers, satellites, cables, and telephone lines, it will be possible to go just about anywhere without having to leave the comfort of your favorite chair. As continually improving technology brings the cost of telecommunication devices down, it will liberate many people from one of the banes of modern society.

The daily commute—so much a part of our lives—could become a thing of the past. In fact, telecommuting has been going on for years in the government and big corporations. Once only the province of the very rich and the very important, the ideas of working at home and teleconferencing are being considered by businesses both large and small.

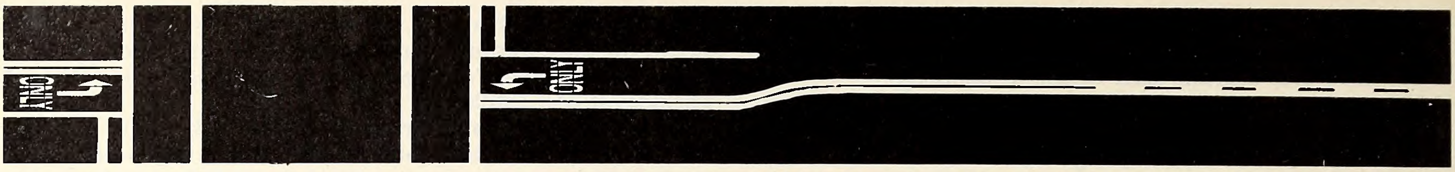
Jack Niles, interdisciplinary studies director for the University of Southern California, is the father of the term *telecommuting*. Niles believes that there could be ten million telecommuters by the year 1990, if certain things happen. His provisos: "The energy crunch must continue; telecommunication networks have to get their act together; and management has to get the message that this will improve productivity."

In January of this year, *Business Week* delved into the potential for telecommuting and came to the conclusion that, although it will never fully replace the traditional office style of working, it will catch on.

The Age of Computer Commuters. Who will actually telecommute—and what jobs are best suited for it—is hard to predict. Not everyone is as optimistic as Niles, and there is some disagreement in the business world as to how much telecommuting will be used. Nonetheless, the equipment requirements are within the grasp of many people. All you need is a smart terminal or microcomputer—even a dumb terminal will sometimes do—a modem, and the proper software. A printer would make the system complete.

At this time, if you had all those things, you could be in touch with many different telecommunication networks and information services. The Source Telecomputer Corporation (McLean, VA), General Telephone's Telenet, and CompuServe (Columbus, OH) are representative of information services scattered all over the country. It's possible to get from the Source everything from tax tips to soap opera news. With the proper equipment you can travel incredible distances at fantastic speeds, capture obscure bits of knowledge, and have them printed up at home by simply lifting a finger now and then.

For some people, the telecommunications revolution will not only make work easier but make quality entertainment easily accessible, too. With all the equipment mentioned above and other devices widely available today, you can turn your study into a library, TV studio, and movie theatre all in one. Christened "media centers" by *Newsweek*, the furnishings of these areas would ideally include a big-screen video format, video recorder, video tape camera, video disk player, top-of-the-line stereo system, film library, and an Apple or some other personal computer. With the technology of three-dimensional holographic projections improving all the time, it may



be possible to dial the atmosphere or environment of your choice. If your favorite restaurant could just deliver fresh Maine lobster and vintage wine over telephone lines or cables, driving to the city for a night on the town might become a thing of the past.

The growing number of communications satellites will make information flow much faster and reach more people than our present means of broadcasting can handle. Multi-beam devices will be able to send and receive millions of messages every second. It's already possible to purchase a microwave dish that intercepts satellite transmissions. With a low-noise amplifier, concave dish, GHz receiver, down converter, and demodulator electronics, you can tap into a Satcom, Westar, or any of a host of other satellites in geosynchronous orbit. You could see first-run movies free, enjoy the best sporting events from all over the world, and keep up on all the latest news from our overseas neighbors.

A Price for Freedom. Some enthusiasts predict that we will soon have wrist communicators like those in the Dick Tracy comic strip. Through cables and video tape, it will become possible to have two-way visual and audio communication. This kind of breakthrough would be a boon to the field of medicine, allowing for long-distance examination of patients by specialists.

Today, the informational freedom of the self-sufficient technologist's vision is not totally free. For instance, if you get your own backyard antenna and tap into the transmission of some cable TV network, you'll probably find your reception scrambled until you pay a fee to the cable company. There's also the prospect of an ingenious breed of computer criminals making life miserable for large companies and, through strict laws enacted to deal with these criminals, for individual users.

The mating of television and microcomputers is right around the corner. Cable TV companies will offer units with microprocessors that allow two-way communication. Theoretically, not only will you be able to change the channel when you don't care for a program, but you'll be able to let the station know just how you feel by punching in on the keyboard, "It stinks."

Telecommuting has been the subject of studies by several large companies in the last few years, including Walgreen's, McDonald's Restaurants, Control Data, and Arthur D. Little. In some, like Continental Illinois Bank and Trust, the effort has been successful; they now use telecommuting as an accepted business practice. On the other hand, some companies that have begun experimenting with telecommuting are not finding it totally acceptable.

Views on Virtue of Vision Vary. The Rand Corporation in Santa Monica, California, has a dozen or so employees—out of a total of five hundred—who work part-time at home on Apples and other small computers. Jerry Koory, Rand's computer services department head, says the at-home effort is not yet fully organized, and that the majority of Rand employees will never telecommute.

"At Rand, telecommuting with personal computers will occur only in certain circumstances," Koory says. Research papers are projects that Koory believes can be done at home, but many tasks, like analyzing a data base, will have to be done in the office.

The problem of monitoring the telecommuter's output—a

major concern for businesses considering this new style of working—is something Koory doesn't worry about. "Are you getting the job done? Things like this can't be measured in hours. It's a matter of whether work got done and, if need be, your supervisor can scan what you've got to see that it did."

Jack Niles's case study of a Los Angeles business and the decentralization of working resulted in a book, *The Telecommunications Transportation Tradeoff* (Wiley, 1976). The good or bad effects of monitoring will "depend on the situation. If you start monitoring invisibly, when does it become invasion of privacy? It will be a while before we know whether these fears are well founded or baseless."

If traditional supervision is removed, what about traditional workmen's compensation? Let's pretend it's a quarter to three in the morning, and you're getting ready to settle down to a few hours of mindless keyboarding. With cup of coffee in hand, you fail to see the issue of *Byte* lying in your path. Coffee, *Byte*, and telecommuter end up in a heap. Unable to keyboard because of a sprained wrist, you wait until morning to call the boss, wondering if he'll ever believe you. Telecommuting may bring about a spectacular resurgence of the tall tale.

Don Burke, senior systems engineer for Telenet, believes telecommuters will have goal-directed jobs. These could be programmers or field analysts who would need no monitoring or supervision, explains Burke. "We have an archaic view of working, mixing output with being in the office nine to five. Still, only a narrow range of jobs, where the boss says 'I don't care how many hours you work as long as *X* and *Y* are done,' will be possible with telecommuting."

Problems to Solve. Burke sees telecommuting demanding a whole new analysis of job functions. Even clerical jobs could be done at home at least in part, including some forms of secretarial work. On the whole, he sees the business world receptive to the notion of telecommuting but believes the practice to be a two-edged sword. There would be a liberating effect—no more driving or riding the trains—but the conditions of management might make it unbearable. Electronic monitoring could reach its dehumanizing worst in cybernated people—human/machine combinations that are little better than extensions of computers.

Microcom (Boston, MA), producer of *Micro-Courier* software system, is one company that's betting on telecommunications to take off in the next couple of years, according to sales director Leonard P. Freed. A veteran of fifteen years in the computer business, Freed believes that the combination of computer power increasing and prices decreasing is making telecommuting—once the stuff of science fiction—a reality. "We have an advertising agency thirty miles away. With *Micro-Courier*, we can send press releases and advertising information, and they can make changes and send it back. We eliminated an hour's drive."

Richard Condon, speaking for TRW, addressed this kind of problem in a recent issue of *Science Digest*. "What sense does it make to take a two hundred-pound man, put him in a four thousand-pound car, and send him to the bank carrying a quarter-ounce check?" he asks.

Freed says it will take "a special kind of individual who is self-motivated and self-directed for telecommuting to be successful. Most people require the positive stroking you get on the job and the social aspects of the work environment."

Long-Distance Collaboration. But teleconferencing is something that Freed sees coming about soon as an accepted business practice. "If you had five general managers in five locations, and they needed to have a meeting, they could just hook up with the network instead of all getting on airplanes and meeting in one place.

"In 1967, when I first started with computers, if anyone had said that I'd have my own computer in the basement by the eighties, I would have said they were crazy," muses Freed. What about fifteen years from now? Microcom expects the telecommunication revolution to change business significantly, though employees working at home may not be a dramatic part of this.

Computer conferencing or teleconferencing is a concept and practice that's been around for many years. Through a central computer, various group members can get together, share information, and make decisions without spending costly time traveling. As telecommunications equipment becomes widespread, educators, doctors, and political leaders will be able to communicate any time with their colleagues, leaving messages if a colleague isn't there. The ability to store messages and other information in computers for later use will eliminate what Freed calls *telephone tag*. This is the century-old game of "I called you and missed you, and then you called me and missed me. And then I called you again. . . ."

Writing is one thing that can be done at home, and many authors have embraced the technological age, giving up their mechanical typewriters for word processers and microcomputers. Larry Niven, Jimmy Carter, Michael Crichton, Alvin Toffler, John Hersey, and Robyn Carr all use computers of some sort for the task of writing.

Through telecommunications it's possible for two authors to collaborate on a book and never see each other. Still, there are some things you can't do with a computer, essential to good relationships between writers; it's difficult for two collaborators to pour each other martinis through their computer terminals, for example.

The Commute to Telecommute Theory. Jack Niles probably knows as much about telecommuting as anyone. As you read this, he might be doing it from his home in Brentwood, California.

Niles envisions widespread telecommuting resulting in local work centers rather than everyone staying at home. This would be like having a small office on Long Island connected to a central computer in the downtown Manhattan office. Via computer, employees who live in the area could conduct work that would ordinarily entail a trip in ungodly automobile traffic or inhumanly crowded rapid transit.

Some people might object to working at home out of concern that the boss might not notice their work and might pass them over for promotions. Some telecommuters have said they would have it both ways—work part of the time at home and part of the time in the office.

Currently, Niles is working on a book concerning microcomputers and telecommuting. "It's mainly a how-to book. This is a modem, and this is how it works. This is a phone line . . . like a manual of sorts." Still untitled, the book should be available sometime next year. The only problem Niles has with Apples is that they come in so many pieces. "What is needed is a nice, integrated machine with all the necessary ingredients for telecommuting."

Niles believes satellite telecommunication will have its uses, though 90 percent of business communication is local. "It doesn't make much sense to send a message through a satellite if it's going only twenty miles away. On the other hand, you could have your office in the mountains, which some people would find advantageous."

The negative aspects of telecommunication and telecommuting may emerge in a loss of the subtleties of personal communication—chiefly those of body language and eye contact. Even with audio and visual two-way communication, some people will not embrace telecommunication, just as some scoffed at the telephone early in the century. Experiments in

telemedicine and telepsychiatry have run into this problem. Psychologists are particularly skeptical; they believe it undesirable to increase the distance between doctor and patient beyond a few feet.

A Society of Hermits? There are also the possible effects on society of having people stay home all the time.

"People don't take the trouble to use their minds anymore. They rely on computers so much they're forgetting how to ask questions." The speaker is Xavier Conroy from John Brunner's classic science fiction novel, *The Jagged Orbit.* In the year 2014, computers have taken over so many human tasks that they now decide who is sane and who isn't. A significant portion of the population is shut up inside massive mental institutions where they interact with doctors only through telecommunications. "We're going insane, our whole blasted species—we're headed for screaming ochlophobia!" mourns the distraught Conroy.

The American Heritage Dictionary defines the noun *ochlophobia* as an "abnormal dread of crowds."

We don't have to end up like this. As long as people still desire one-on-one meetings, there is nothing to fear. Telecommuting could have very positive effects on human existence. The reduced time spent traveling will leave time for other activities, such as recreation and social interaction.

Telecommunications can put you in touch with many people you might not otherwise have known about. Just imagine all the people you've met over the telephone that you'd never have had a chance to meet if you had had to rely on physical proximity. Through bulletin boards and information networks, you can locate people with your particular interests with whom to share ideas and information.

Telecommuting will open many opportunities to handicapped and shut-in individuals. And parents who would choose to stay at home with their young children will be able to do so and still hold down part-time or full-time jobs. Instead of shutting off the world by keeping the individual at home, telecommunications could bring real independence.

Try It—You'll Love It. The desire to roam as far as one can and experience all that life offers is something Americans have always had. Much more obviously than space exploration, the telecommunication revolution will make the wild, new frontier of the information age available to many people. It still takes courage and a strong will to take the first steps, but the reward is undeniable. In the next two or three decades, we'll see a lot of people leaving the piped-in garbage from the idiot box for the far more challenging and rewarding task of learning about—and contributing to—the world we live in.

No more flat tires, no more long gas lines, no more traffic jams, no more speeding tickets, and no more downtrodden hitchhikers. This is what the future holds for the self-motivated worker who can commute by telecommunications.

Woody Guthrie roamed that ribboned highway. Our future poets and songwriters may wander the telecommunications skyway. ■

ARTSCI explains why some word processing systems are better than others.

Let's begin with an easy to understand explanation of what a word processor is and how ARTSCI has created a professional system.

A word processing system is simply an easier, faster and less expensive way to type. With a modern word processor, documents are entered on a video screen instead of paper.

You can enter your first rough draft without concern about errors or spelling. Simply go back and insert letters, delete words and even move paragraphs with a few keystrokes. No document will ever have to be retyped.

WORD PROCESSING AND THE APPLE II

The APPLE II is the most expandable, inexpensive micro-computer available today. It can perform almost any task, including word processing.

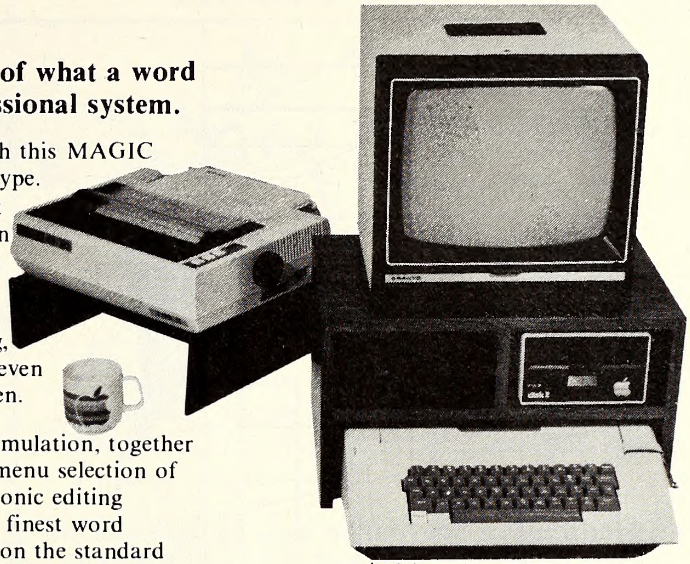
The standard APPLE II however, uses a 40 column video display. This display causes a serious word processing problem: How do you display a full sized 80 column letter? Most word processing programs available today do not solve this problem.

THE MAGIC WINDOW

ARTSCI has developed the MAGIC WINDOW word processing system that incorporates the full power of a professional word processor and solves the APPLE'S display problem without expensive hardware.

The first feature of a professional word processing system is the ability to enter and edit data in a fast and friendly manner. The MAGIC WINDOW operates just like a standard typewriter. The electronic paper moves to the left across the video screen as you type. Almost any size document can be represented on the video screen. You can see the edges of the paper through this MAGIC WINDOW as you type.

The rule is: What you see on the screen is what you'll get in print. However, if you print using proportional spacing, the result will look even better than the screen.



Apple is a trademark of Apple Computer Inc.

This typewriter simulation, together with simple to use menu selection of functions and electronic editing abilities, creates the finest word processor available on the standard APPLE II.

MAGIC SPELL

The second feature of an advanced word processor is the ability to find and correct mistakes. The most common mistakes in most documents is the misspelled word.

ARTSCI's MAGIC SPELL program will take any document you can create and find spelling errors. Over 10,000 commonly misspelled words are known to MAGIC SPELL. You can also add new words to the vocabulary at any time.

CUSTOM LETTERS

The third feature of a professional word processing system is the ability to alter a document by replacing names and other related data from mailing lists.

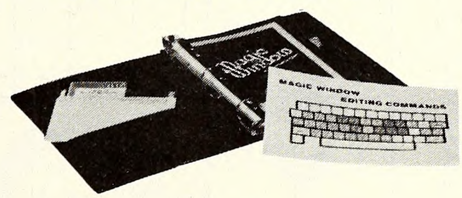
Form letters, invoices, and almost any document can be individualized by replacing names, addresses or any other personal data anywhere in the document using ARTSCI'S BASIC MAILER.

By using the BASIC MAILER you can take any mailing list and sort through the list by different criteria and print personalized letters with a few simple keystrokes.

A COMPLETE SYSTEM

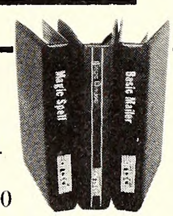
These three programs, THE MAGIC WINDOW, MAGIC SPELL AND THE BASIC MAILER, together form the only complete and professional word processing system available on the standard APPLE II computer system.

ARTSCI TAKES THE WORK OUT OF WORD PROCESSING



For a more thorough; explanation of the ARTSCI word processing system send for our free booklet.

artsci INC



Mail to:
ARTSCI INC.
10432 Burbank Blvd.
North Hollywood,
California 91601-9990
(213) 985-2922

Name ______________________

Company ______________________

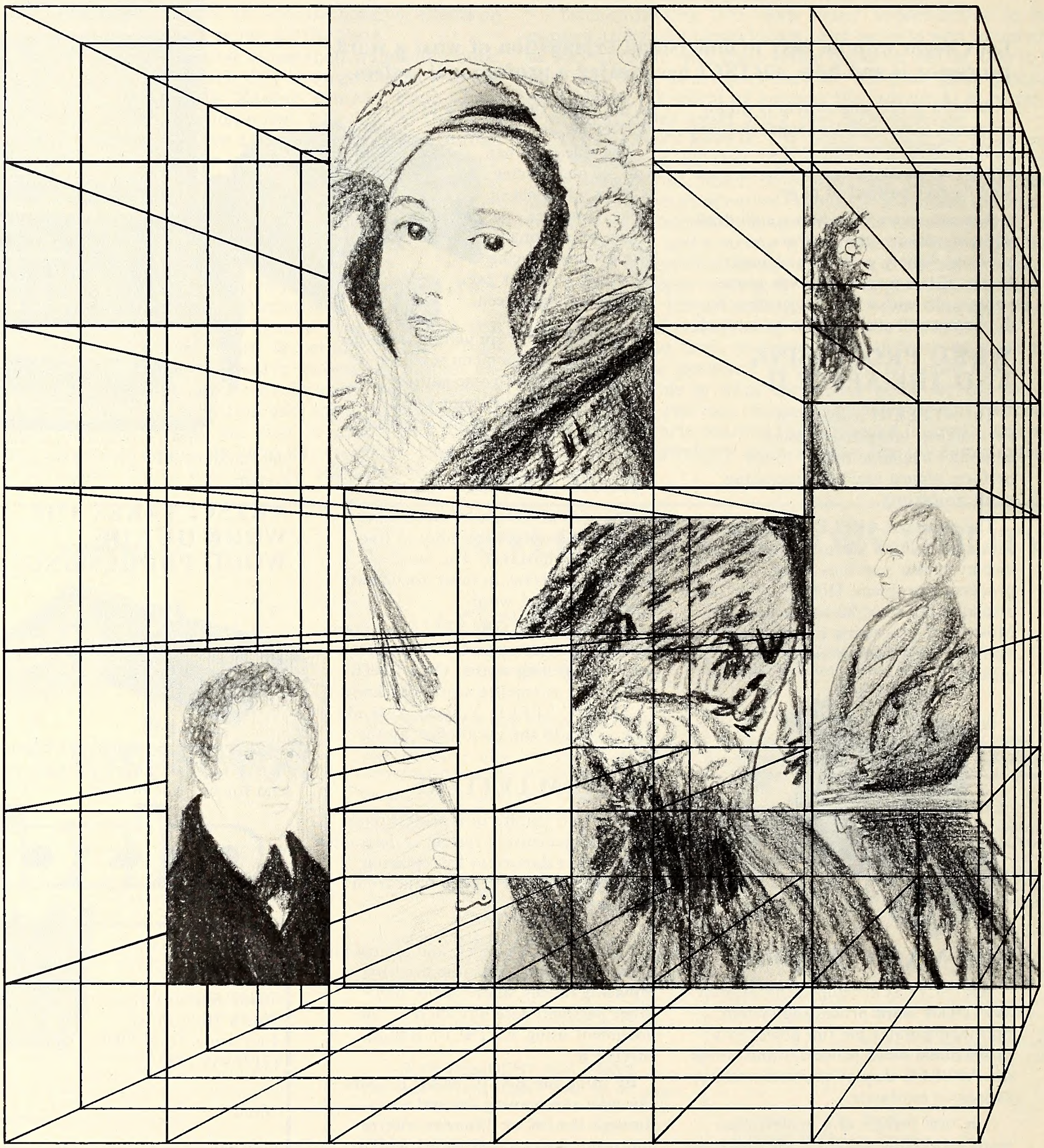
Address ______________________

City __________ State ________ Zip ____

Phone ______________________

ST PE MI CC BY CA

ADA



Nineteenth Century Woman, Twenty-First Century Mind

BY CAROL JAMES

It might have been just another social gathering. Ada, a shy girl of seventeen, was still self-conscious at parties; an adolescence characterized by ill health and seclusion at home had hardly prepared her to take part in adult party conversation.

But there was one person at the party to whom she was drawn. He was a middle-aged widower, not particularly handsome, but he fascinated her. Introduced to her and her mother—Lady Annabella Byron—as a celebrity, he was Charles Babbage, inventor of the internationally acclaimed difference engine—the first calculating machine.

Ada had never met anyone quite as brilliant as he, though her private tutor was William Frend, a famous mathematician. Frend had taught her to love astronomy and enjoy algebra and geometry as recreation; but Babbage's contributions to science and mathematics were truly awe-inspiring.

Unladylike Interests. Ada was thrilled when Lady Byron, also intrigued by the idea of Babbage's difference engine, promised her they would soon go to see the invention for themselves. Both were enthralled when they saw it, but it was Ada who began attending lectures on the machine's workings and actually grasped the mathematical principles involved. Perhaps she would be a scientist herself someday, she thought.

Had Ada come of age in this day rather than in the early days of Queen Victoria, she might have realized more fully the career of which she dreamed; but the circumstances of her life decreed otherwise.

Her father was the poet Lord Byron. Shortly after Ada's birth in London in 1815, her parents separated because of suspected incest between Byron and his half-sister. Byron, in any case, was an incurable wanderer and notorious womanizer; for that reason, Lady Byron never allowed father and daughter to meet again. Nevertheless, Byron never forgot Ada and periodically begged for news and pictures of her—usually in vain—for the rest of his life.

As a child, Ada was described by her mother as lively and imaginative, with a talent for learning and unusual powers of observation. She also showed ingenuity in designing and building ship models and other mechanical toys; and when her mother took her on a tour of England's industrial cities to see the manufacture of ribbons, pottery, and cutlery, Ada was fascinated by the machinery.

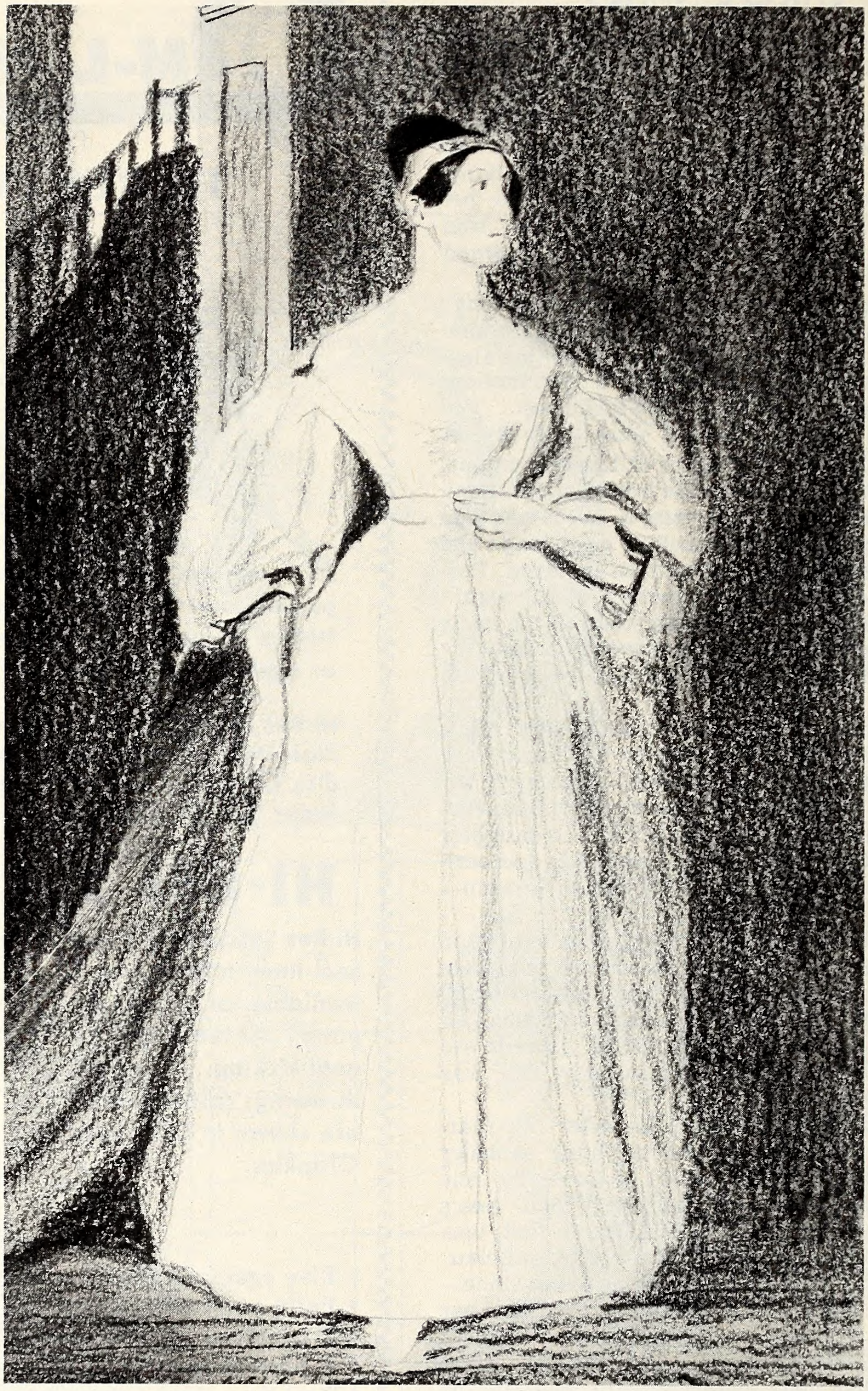
Two Men Who Never Heard of Chauvin. After her marriage at nineteen to an amiable, attractive, young bachelor named William King—later the first Earl of Lovelace—she continued to study mathematics with William Frend. Ada's last afternoon before the birth of her first child, in fact, was spent doing math problems with the elderly mathematician.

The young wife—now known as the Countess of Lovelace—was fortunate in that her husband and mother encouraged her scientific interests in a day when most women would have been far more restricted. However, frequent ill health, social obligations, and care of her three children did hamper her, so that at times her scientific work was limited to keeping a scrapbook of entertaining math problems.

Despite her handicaps, she was determined, by the age of twenty-three, to become a scientist. She repeatedly wrote letters to Babbage, asking him to be her mentor. She didn't wish to appear conceited, she said, but she was sure she had genius and could go as far as she liked in mathematics and various scientific pursuits if he would help her. In return, she and her husband promised financial support for his inventions, as well as use of their considerable influence to get government subsidies for him.

Talent Hidden in Satin and Lace. Unfortunately, Babbage was too busy to do more than visit them occasionally and comment on the letters Ada wrote him, letters filled with scientific speculations, mathematical problems, and formulas.

She published papers when time and health permitted, boldly sending copies to prominent scientists of the day. She also studied their work and sometimes dared to correct their



portrait of Ada, Countess of Lovelace, aged 18.

math errors. But Babbage, whose full appreciation of her talents she most avidly craved, continued to treat her as a clever amateur rather than a colleague.

Friends saw her as flighty, with fairylike beauty and mysterious charm; but she was frustrated by her limited professional opportunities and often depressed—a condition made worse by occasional not-so-secret drinking bouts and experimentation with opium-based drugs.

At twenty-seven she read Babbage's treatise on his revolutionary new digital computer invention, which he called the analytical engine. Developed in the 1840s, the machine was the original precursor of our computers.

Babbage had written the treatise in French to be read at a scientific meeting abroad. Ada decided to translate the work into English along with another French-language treatise on Babbage's invention by the French scientist General Menebrea.

Babbage's Awakening. When she sent the finished translations to Babbage, he was flattered and then suggested she add a few notes of her own. The notes took two years to write and were much longer than the original work. They are said to show a complete understanding of computer programming principles a century before they were put into practice.

The delighted Babbage, upon reading them, wrote back, ". . . the more I read your notes the more surprised I am at them." He had finally discovered the abilities in Ada that she so wanted him to recognize, and he was charmed. In typical Victorian fashion, he romanticized her, calling her his "enchantress of mathematics." Having spent years trying to

The A2-GE1 Graphics Editor

for the Apple II

You bought your Apple for its graphics capabilities. Now with the **A2-GE1**, you can use those capabilities to the fullest.

With **Object Editor** you can create whatever objects you want in the colors of your choice. You can also type in whatever 3D text you want and in different sizes. And saving an object is as easy as naming it.

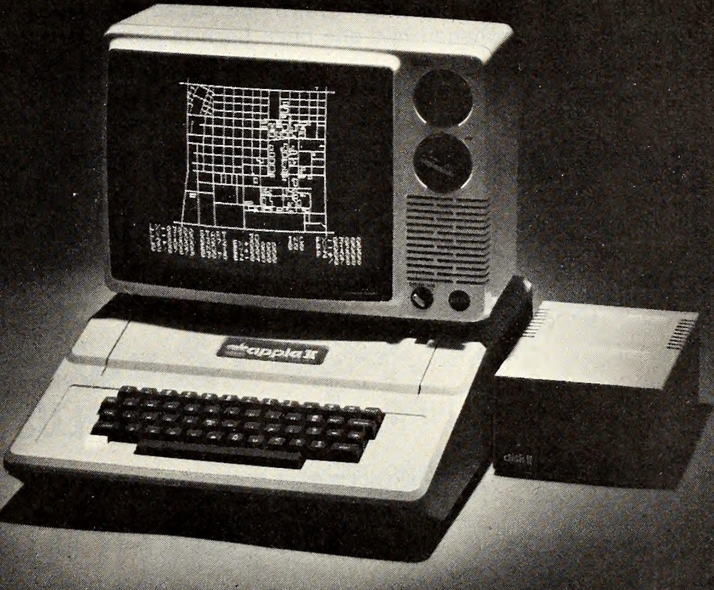
Then give the object names to **Motion Programmer** and see how the beautifully laid out keyboard controls will let you switch objects on or off, animate them, or add upper or lower case 2D text mixed right in. It's remarkably easy.

You can also record your entire presentation, animation and all, for later use with **Motion Playback**, or just take "computer snapshots" of scenes with **Slide Show Playback**.

The **A2-GE1 Graphics Editor** requires the A2-3D1 or 3D2 and includes **Object Editor**, **Motion Programmer**, **Motion Playback**, and **Slide Show Playback**. It also includes a special A2-3D2 interface for BASIC programmers.

See your dealer for a demonstration.

Graphics power for the non-programmer!



Map of the University of Illinois campus being constructed on the Apple II.

A2-GE1 Graphics Editor
$34.95 on disk (48K and A2-3D1 required)

A2-3D1 Graphics Package
$59.95 on disk (32K required)

A2-3D2 Enhancement
for color and independent object manipulation
$24.95 on disk (48K and A2-3D1 required)

A2-3D/A Saturn Navigator
a 3D adventure program
$24.95 on disk (48K, A2-3D1, and Applesoft in ROM required)

subLOGIC
Communications Corp.
713 Edgebrook Drive
Champaign, IL 61820
(217) 359-8482
Telex: 206995

For direct order, include $3 for UPS or $5 for first class mail delivery. Illinois residents add 5% sales tax. Visa and Master Card accepted.

get the inventor to take her ambitions seriously, Ada must have derived deep satisfaction from his comments.

Ada's proud husband encouraged her to sign her own initials to the Menebrea treatise when it was published. The general was curious to know who among his fellow scientists possessed the initials "A.A.L." When he discovered they belonged to Augusta Ada, Countess of Lovelace, he was amazed—not only at the accuracy of the translation, but by the commentaries she had added. It was this project—largely unknown to all but a select few in Ada's own time—that has won her; in the twentieth century, the title of first computer programmer.

Time Out of Joint. Her next endeavor was to devise the perfect system for betting and winning on racehorses. She and Babbage—with Lord Lovelace as assistant—carried on a secret correspondence to this end, with Babbage now describing himself as her slave, and she assuring him she was "working very hard for you; like the Devil in fact." Well-meaning friends feared her intense studies would undermine her physical and mental well-being. In fact, her health—always delicate—did begin to fail about that time, for she was suffering from the cancer that would eventually kill her.

Horse race gambling became a disastrous addiction. By age thirty-four, she was heavily in debt, pawning family jewels behind her husband's back and secretly borrowing money from her mother to pay the bills.

Her last two years were spent in mental anguish over her financial difficulties as well as physical torment. She died in 1852 at the age of thirty-six.

Her story is one of tragic proportions, since the contributions she made to science were small compared to what she might have accomplished had she been born in our time. Her chronic health problems—which seem to have been her greatest career handicap—would surely have been more effectively treated today. It would be a great shame if a female mathematician didn't find it substantially easier to gain recognition among her peers today than Ada in the 1840s.

The woman who lived as much in the future as in the present would surely be gratified to know that she has been chosen to symbolize a new technology in the 1980s.

Posthumous Recognition—and a Controversy. Recently, the United States Department of Defense adopted a specially developed programming language called Ada, after the Countess of Lovelace, to control all its embedded computers in planes, missiles, and submarines. Experts predict that the language will soon have wide influence throughout the computer and microcomputer world.

Based on Pascal and PL/1, Ada incorporates some of the most sophisicated software methodology of the past decade. Its features include modularity and data abstraction; strong typing; separate—but not independent—compilation; tasking and distributed processing; exception handling; and generic units.

The language, expected by the defense department to be fully operational by 1984, is not without its critics. Its very complexity and efficiency, some say, may lack enough of those human-intervention safeguards that any military device must have to ensure against nuclear accident. Ada's defenders, on the other hand, point especially to its exception handling feature as a built-in emergency system.

One wonders what the countess herself would have thought of the controversy surrounding the language that bears her name. It's known that she often cautioned others against putting too much faith in computer systems; the human element must always be present, she warned. As she wrote to Babbage in 1845 concerning the analytical engine: "No one knows what . . . awful energy and power lie yet undeveloped in that wiry little system of mind."

FEELING THE SQUEEZE?

THE HOME ACCOUNTANT

can make your budget fit!

• 100 User defined budget categories • Up to five checkbooks maintained • Hi-Res graphing of budget uses actual and trends • Personal Balance Sheet and Income & Expense Summary with comparison to budget or previous month* • Unlimited annual transactions • Fiscal or calendar year • No printer required • Optional check printing* • All transactions may be edited at any time • On screen editing during bank reconciliations • Automatic checkbook transactions • One program for checkbook, cash and credit card transactions • Transactions may be flagged for tax purposes • Transactions may be "split" among more than one budget category.

NEW

from

CONTINENTAL SOFTWARE

12101 Jefferson Blvd., Culver City, CA 90230 • (213)371-5612

Introductory Price $74.95

*Printer required for these options. Apple][is a registered trademark of Apple Computer Inc.
Hardware Requirements: Disk drive, DOS 3.3, APPLE][48K, Applesoft in ROM. Printer and second disk drive optional.

Oo-Topos, the first and only adventure program written directly for Microcomputers by a science fiction writer. Now, Michael Berlyn, author of Crystal Phoenix and Integrated Man transports you to the truly distant, alien world of Oo-Topos. It's an original science fiction tale programmed for adventure.

See your local software dealer and discover Oo-Topos.

Requires DOS 3.3 Applesoft* in ROM

From

SENTiENT SOFTWARE

P.O. Box 4929, Aspen, Colorado 81612

(303) 925-9293

©1981 by Sentient Software, Inc.

"...deserves a top spot in the realm of adventuring."
— Softalk Magazine

Exclusively distributed by

SOFTSEL

8295 S. La Cienega Blvd.
Inglewood, CA 90301
(800) 421-5770
In CA (800) 262-4242

*Applesoft is a registered trademark of Apple Computers, Inc.

The New Compilers— WHO NEEDS 'EM?

by ROGER WAGNER

Within the last six months, several software products generally described as compilers have appeared upon the market, and they seem to have caused something of a commotion. There also exists a certain degree of confusion about exactly what a compiler is and what the average user can expect out of using such a program.

Also, and perhaps of even greater importance, I'd like to discuss some of the things compilers *cannot* do. You will then presumably be in a much better position to judge for yourself the value of such a utility to your programming goals.

Basic Versus Machine Language. Since you own an Apple, you are at least mildly familiar with Applesoft Basic. It is a relatively straightforward programming language that uses commands not unlike everyday English. If you want the computer to print a name, you simply enter PRINT "NAME" as a command statement. Although this works, there is quite a bit of behind-the-scenes activity going on to make the end result happen. The 6502 microprocessor does not directly understand the command *print*, so some kind of translator must be provided.

What the 6502 does understand is something commonly referred to as machine language. In machine language, various number values are placed in a series throughout a given range of memory. As the 6502 scans through a given range, it executes extremely simple actions based on the values found there. These typically are limited to simple addition or subtraction and a limited number of other rather trivial operations. More complex functions are made up by assembling groups of these simple commands into larger routines.

Each approach has its associated benefits and problems. Applesoft Basic is much easier for the beginner to learn and generally takes less time to create a given program. Machine language, on the other hand, is extremely fast in its operation and tends to be more compact. Typically, a machine language program may take up only half the room of an equivalent Basic program, and execute a hundred times as fast!

That brings us back to the translator mentioned a moment ago. There are actually two different kinds.

Interpreters Versus Compilers. Normal Applesoft (and Integer Basic for that matter) uses an interpreter to run a completed program. The interpreter, always present in the computer, is located at the very upper end of memory at addresses \$D000 through \$F7FF.

Don't be concerned if the numbers don't mean a great deal to you. The main point is to understand that the interpreter is itself a set of machine language routines that reside either in the ROM chips on the motherboard of the computer or on a language card of some sort in slot 0.

When a Basic program is run, the interpreter scans each line, byte by byte. When a statement such as *print* is encountered, two things happen. First the command is looked up in a table within the interpreter. Then a jump to the appropriate subroutine (also within the interpreter) is done. In the case of *print*, the subroutine would digest the characters following the *print* (if any) and output them to the screen (or printer, disk, and so on) as commanded.

This constant translation is the main reason for the slow execution speed of Basic. Until recently, an interpreter was the only option for an Apple Basic program.

A *compiler*, on the other hand, is an ideal-sounding utility that takes a source Basic program and converts it into the machine language equivalent to perform the same task. How marvelous! We can now write the program in Basic (easy), compile it, end up with a file half the size that runs a hundred times faster, and even throw out the interpreter to gain some

extra room in memory. After all, pure machine language should no longer require the interpreter to function, right?

Alas, ours is not such an ideal world. The resulting code is now twice as large as the source program, only runs two to five times faster and still requires the interpreter to be present. Let's see why.

How Compilers Work. When the print statement is coded into the Basic program, the computer does a very clever thing. Rather than store the letters P-R-I-N-T, a single byte is stored with a value that signifies the *print* command. This not only saves memory, but also helps speed up the already slow table look-up process. (Now it only has to look up a one-byte entry as opposed to an entire string.)

When the program is compiled, rather than write an entirely new equivalent machine language program, the compiler simply substitutes things like *print* with jumps to routines that would have taken place if the program were run normally. In effect, the compiler stores only what would have been done with the command that would have done it. Many times, this still amounts to calls within the Applesoft interpreter, so its presence is still required.

In memory, what used to be a one-byte *print* command now becomes a three-byte jump to a subroutine: one byte for the JSR command, two bytes for the address to jump to.

This would seem to imply a three-to-one expansion of the source code. However, the characters following the print are kept in the compiled code as they were, so they create no expansion; thus, they compile on a one-to-one ratio.

The net result is about two to one overall, varying according to the program being compiled. Also, some additional routines (usually called the run-time package) are required in the compiled code. These handle general maintenance such as clearing all variables to zero when the program is run.

A speed increase results because very little needs to be looked up anymore. Everything tends to be done as absolute jumps to fixed points in memory.

An excellent example of this is a *goto* statement. In Basic, whenever a statement such as GOTO 150 is encountered, the interpreter must stop to search the entire program to find line 150. It then jumps to that address in memory and continues there. In a compiled program the *goto* is replaced with an absolute jump to the place in memory where the compiled equivalent of line 150 ended up. This eliminates *two* searches: the search to translate the command GOTO 150 and the search to find the line in question. Instead the jump is immediately executed. Total time is a few microseconds as opposed to a few hundred microseconds the other way.

Limitations of Compilers. After all this, there are some other things that keep us from feeding every Applesoft program we have to a compiler. These are generally matters either of size or of unsupported commands.

If we assume a two-to-one expansion factor and about 32K of available memory at compile time (remember, the compiler needs some room for itself), then the largest program that can be compiled would be 16K—about sixty-six disk sectors. This is still fairly large, but not as large as the biggest Applesoft program that could be put in memory.

The second constraint varies among compilers, but in general amounts to the omission of some legal Applesoft commands. It's rather obvious why some aren't provided for. *List* and *del* relate to line numbers that aren't present in a compiled program. Tape commands are usually considered obsolete, so *save, load, store,* and *recall* are usually not allowed. Whether a compiler will recognize *lomem*:, *himem*:, *resume, &, usr*(), and *cont* varies from product to product. You'll have to do some investigation to see which products support which commands.

Of particular interest are the ampersand and *Dim* statements. Ampersand can be easily supported in terms of doing the proper jump when the *&* is encountered. The problem is that most ampersand routines are written to process standard Applesoft variables. Since these no longer exist, it is *very* unlikely that the routine will function properly. This is not a fault of the compiler; rather, it's an unavoidable result of the nature of the beast.

Similarly, Applesoft allows a statement like DIM A$(N) where *N* has been given some value. None of the presently available compilers allows this type of statement. Instead you must use a constant, such as DIM A$(50). This is another direct result of the nature of the compiler, in that variable space is set up *before* running in compiled code.

What To Feed Your Compiler. Not *VisiCalc,* not *Raster Blaster*. Do not expect to be able to compile arbitrarily every program you own. But most compilers are designed to digest the majority of standard Applesoft programs with a minimum of difficulty. They are written to make your programming life easier.

In terms of where to expect the greatest speed increases, *goto* and *gosub* statements, logic testing (*if-then,* and so on), and *for-next* loops do terrifically well. That's where the ten-to-twenty-time speed increases come in. Operations (mostly math) that still rely on the Applesoft routines in the interpreter, such as *hplot* and *sin(X)*, will not do as well, although some increase will still be apparent.

Why Compilers Are Valuable. After all this, you may be thinking, "So why use a compiler at all?" The first reason has already been mentioned: the speed increase. When writing a program that has an exceptionally slow routine in it, any speed increase is desirable. An increase of two to five times is not insignificant. What used to take an hour might now be done in as few as twelve minutes. In some cases, the speed increase can be even greater.

The second reason has to do with program security. This aspect is often overlooked but can be more important than the speed issue. If you have any portion of a program you don't want other people to examine, compiling makes the code virtually inaccessible. This might be a concern if you have proprietary algorithms or security related concerns such as passwords, pay scales, and so on. In addition, some compilers dis-

cuss how to make a program immune to control-C and reset, so that it's impossible to stop the program once it's running.

In many instances, these two reasons more than compensate for the limitations of using a compiler. Many programs that were previously impractical become very desirable with the benefit of power.

How To Use a Compiler. Use your compiler with good judgment. As with any tool, its usefulness is a direct result of its own inherent features combined with your knowledge of how to use it best. In the case of compilers, a wide variety of options given at compile time directly affects how much control you have over the completed code. You should be aware of which ones offer the options you need to suit your own programming style.

You *will not* be able to compile blindly every program you've got and get a miraculously compacted module that shoots smoke out the vents due to its phenomenal execution speed. You *will* be able to speed up certain programs by a noticeable amount and increase the listing's internal security.

In fact, maximum power is gained from a compiler when you write the original code with eventual compiling in mind, thus taking advantage of the greatest number of features in a given compiler. This is because most compilers allow for the addition of internal *rem* statements that actually add to the power of the Applesoft program. For instance, for a very large program, it may be possible to jump over the area of memory used by the hi-res page, something that is virtually impossible to do in a normal Applesoft program. Also, you may be able to have several programs in memory at once, each of which can call another. It's even possible to have an Applesoft program call a compiled subroutine, passing its variables to operate on and retrieving certain results. Some of these features may not be available on all compilers, so some shopping is recommended.

The general guidelines are as follows:

1. Because of the expansion factor, it is best to use a compiler where you want a speed increase, and the programs are either short enough that size is not a concern or can be chained by using common variables.

2. Where size is a concern and you don't wish to chain modules together, you may wish to compile just a portion of the program and call that subroutine from a master program. This restricts expansion to the part of the program that can take best advantage of compilation.

3. In cases where security is a concern, compilers can be quite useful when used in certain ways. By combining the control-C and *reset* traps, you obtain a very reliable package that's nearly impossible to get out of except through the procedures you allow. Also, unique mathematical algorithms can be protected from user examination or modification since compiled code is difficult to decipher even to an experienced machine language programmer. Finally, portions of a program containing such things as passwords and pay scales can be concealed from examination by compiling.

Note that using a compiler to secure algorithms or passwords requires only a portion of the program to be compiled, leaving the rest of the listing open to modification.

How To Choose a Compiler. With a number of compilers to choose from, look carefully at what *you* need. See your local dealer and watch for reviews in user magazines. For best results, you'll want as much control over the final code as possible, so look for the largest number of compile-time options within a compiler that meets your other needs. Make sure the package includes an understandable manual that will allow you to get the maximum results from the package. All that a program can do is insignificant if you can't understand how to make it do them.

In general, look for the best match between what you need for your programming style and goals and what a compiler offers. Think about what kinds of things you're likely to need now *and* in the future.

By carefully selecting the best package for your particular needs, you should end up with a programming tool that's sure to be a valuable aid now and for some years to come.

FIND THE MISSING ALKEMSTONE™
AND CHANGE YOUR DREAMS INTO REALITY

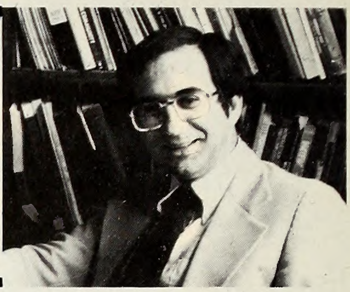
Legend says the ALKEMSTONE will grant its owner anything he desires.
We at Level-10 have been searching for the Alkemstone but we need your help.
All of the clues are contained in the Level-10 Alkemstone game.
You decipher them and discover where the stone is hiding, and we will reward you with $5,000.
And here's your first clue: see your nearest Apple dealer.

LEVEL-10™

A DIVISION OF DAKIN5 CORPORATION
P. O. Box 21187 • Denver, CO 80221 • (303) 426-6090

Mind Your Business

BY PETER OLIVIERI



This time of year, many of you are looking ahead to the holidays. Thanksgiving is tomorrow, and Santa is not far behind.

For many businesses, now is also the time to think about closing the year. This usually means putting the books in order, perhaps taking an inventory, and maybe even preparing for a formal audit of the records. Soon after these tasks have been taken care of, it will be time to assemble the documentation our Uncle Sam requires and pay any taxes that are due.

As we've begun to illustrate in this column, the computer can be an enormous aid to business owners and managers in terms of maintaining records, providing helpful information, and supplying appropriate documentation of what has been going on in the business during the past year.

The last few columns have focused on data-base management systems—packages that allow you to create a data base and interact with it. Thus, you might create an inventory file, personnel file, vendor address file, customer file, or any other data file appropriate to that particular business. These general packages leave much of the application design to you. A DBMS package is a must for the serious business computer user.

A host of other programs have been designed specifically for business. Since businesses often rise or fall on their success at managing financial records, software for these functions are among the more common business applications.

Financial Records. Most of these applications fall under the category of accounting applications. This category includes programs that maintain a general ledger, accounts payable, accounts receivable, payroll, and inventory control. Such programs provide a means of recording events that happen to a business over a period of time. Before looking at what some of these packages will do, let's consider some accounting terms.

Every business has certain transactions that take place daily. These transactions affect the firm's *assets*—things of some value—and *liabilities*—things owed to others. One of the reports a business maintains is a *balance sheet*, on which the status of its assets and liabilities is reported. Another report, the income statement, accumulates daily transactions over a period of time. Since these two reports are important in analyzing the financial health of a business, they deserve to be examined in a little more detail.

While the exact content of each of these reports varies according to the size and complexity of a firm, most firms' reports include the following items.

The Balance Sheet. The format of a balance sheet has evolved over many years and now is pretty well standardized. The typical balance sheet records assets and liabilities at a particular moment—usually the moment the sheet was prepared. The total assets and liabilities recorded must balance—hence the term balance sheet.

Assets on the balance sheet are usually divided into three major categories—current assets, fixed assets, and other assets.

Current assets consist of cash—or things that can be converted into cash—in a reasonable amount of time. These may include securities that can be converted easily, accounts receivable (money owed to the business), and some inventory items. *Fixed assets* are usually represented by equipment. The category *other assets* would include long-term investments, patents, or copyrights that are owned.

Liabilities on a balance sheet are also divided into three categories. These are current liabilities, long-term liabilities, and shareholders' equity.

Current liabilities are obligations expected to come due in the current operating year. These would include accounts payable (debts the firm owes), notes payable (slightly longer-term debts), and certain liabilities that have accumulated over the year (such as taxes and wages). *Long-term liabilities* are those with a due date more than a year away. These may be loans from a bank. The final category, *shareholders' equity*, represents claims the owners have on the business. These may take the form of common stock (which may have been issued to obtain funds to start or promote the growth of the business) or earnings that have been put back into the firm.

The Income Statement. When investors are considering whether to invest in a firm, they want to know more about it than a simple examination of its balance sheet will reveal. They're also interested in the growth potential of the business.

Your Direct Line to Computer Gaming—

SOFTLINE

You'll fall hook, line, and sinker for this new magazine that spans the world of computer gaming in the inimitable Softalk fashion.

Features on new games, tutorials on new techniques, interviews with prominent game-makers, a look at coin-operated games, discussions of what constitute good gaming values—all these are part of Softline.

You'll also be able to get a line on your progress in your favorite game relative to other gamers. Can you top Woz's score in Sabotage? Is your Gorgon score or your Apple Panic score competitive? Will you be the first one on your block to have your name in bold print for a high score in Space Quarks?

Softline panders to the frivolous in most of us. If you indulge in computerized entertainment, you should sign up for Softline.

Softline is distributed free bi-monthly to all owners of personal computers who have an interest in computer games. Send your name, full mailing address, make, model and serial number of your personal computer to Softline Circulation, 11021 Magnolia Blvd., North Hollywood, CA 91601.

For this reason, the income statement becomes an important instrument to examine.

A typical income statement contains the following items:

1. *Revenue* reflects the amounts received (or due to be received) by the firm over a specified period of time.

2. *Cost of goods sold* is deducted from the revenue amount. It represents the expense incurred by the business in generating revenue.

3. *Gross profit* is the remainder after subtracting the cost of goods sold entry from the revenue entry. This represents the amount of money that remains for running the business.

4. *Operating expenses* are deducted from the gross profit. Among these expenses are the costs of advertising, insurance, administration, heat, and light.

5. *Profit* is what remains after operating expenses have been deducted. Profit is a sum of money that can be paid to owners or spent for expansion and other things.

6. *Income taxes* constitute one of the other things that decrease a firm's profit.

7. *Net income* is the final item on the income statement and represents what is left after income tax obligations have been accounted for. Remember, the income statement reflects the operations of a business over a period of time.

So what? You may be wondering what all of this has to do with computers. Well, if a computer can help you keep track of all these items and produce some of these reports for you, then it's certainly a valuable resource. (The computer itself may be an entry on your balance sheet, representing an asset.) But let's go a bit further before getting to the computer as a helper in all this.

General Ledger. All items appearing on a balance sheet or income statement are reflected in individual accounts. When these are grouped together, the document that results is called a general ledger. (In the old days, all accounts were maintained in bound ledger books—hence, the name.) These ledgers are the same as the files we've been discussing all along. There may be ledgers to cover accounts receivable, raw materials, inventory items, work in process, accounts payable, and so on. Thus the general ledger represents the financial records of the firm. Such detailed record keeping can become quite complex, even for small businesses.

In summary, then, a firm's accounting activities include processing transactions for various accounts, entering each business transaction in a journal, posting the information that has been recorded in the journal to the accounts in the ledger, adjusting the accounts as necessary, closing the accounts for a particular accounting period, and producing the required financial statements.

Enter the Computer. There are several software packages that can be used by a business to maintain its financial records. We'll describe some of them briefly here and review them in depth in the next issue. Two factors in particular are important in selecting a business software package. First, the

programs in the package should talk to one another; that is, information from one program should be available to the other programs. This makes it possible to create a general ledger from subsidiary ledgers. Second, each program should be able to work alone, since some organizations may not wish to use all a package's options.

The Controller. *The Controller* is a package produced by Dakin5 (Denver, CO), distributed exclusively by Apple Computer. More than seven thousand copies of the package have been sold. The package contains a set of programs that act as an automated accounting system and can handle accounts receivable, accounts payable, and the general ledger.

Dakin5 has also released a companion product, *The Analyzer*. This package uses *The Controller* as a data base and produces sixteen additional reports.

The Controller presents a list of options (menus) that do not presume extensive accounting knowledge on the part of the user. Some of its features include: cash management, identification of slow paying customers, tracking accounts payable by due date, automatic posting and updating, summary reports for management, detailed reports for analysis, vendor and customer lists, mailing labels, individual account statements, extensive error checking, and historical record keeping.

Accounts Receivable. *The Controller*'s accounts receivable system can be used independently or in conjunction with its general ledger system. Up to 250 customers per diskette and a maximum of one thousand transactions (for all customers) per month per diskette are possible. As many as nine diskettes can be used with the accounts receivable system; a minimum of three is recommended.

You may have up to one hundred salespersons. The largest amount allowed for one sale is $99,999.99, and the limit for one tax amount is $999.99. This program can prepare up to 250 statements per diskette per month. The largest single accounts receivable balance for any one customer per aging period is $99,999.99. The largest overall balance allowable for any customer is $399,999.96. Sales entries and payment entries can be entered in batches as large as one hundred. The total batch amount for sales and payment entries is $9,999,999.99.

Up to one hundred transfer journal entries per diskette can be made to the general ledger system from the accounts receivable program at any one time.

Accounts Payable. The accounts payable program also works either independently or in conjunction with the general ledger. This program can accommodate up to 125 vendors at any one time. The total for all accounts payable for any one month can be as large as $99,999,999.99. You may print as many as 300 checks each month and process as many as 300 invoices per month. These may be entered in batches of 100 invoices at a time subject to a batch maximum of $99,999,999.99. Up to 300 unpaid invoices may be in the system at one time. Again, up to 100 transfer journal entries may be made to the general ledger at a time.

General Ledger. The general ledger system can handle up to 250 accounts, with a maximum balance of $99,999,999.99 per account. You may use *The Controller*'s own chart of accounts or customize your own. The chart may be divided into ten different departments if you wish. The system can handle up to 1,000 journal entry transactions per month and will accept up to 250 transfer entries at any one time.

Processing Speeds. For very large files, both the Apple and *The Controller* slow down considerably. While the actual limits to the system are quite high, the developers suggest that it's more practical to stay within the limits shown in column two.

	Actual	Recommended
A/R transactions per month per disk	1000	750
G/L transactions per month	1000	750
G/L journal entries per month	1000	700
G/L batch entries	250	100

The user's guide that comes with *The Controller* is nicely done. It has been designed both for the novice and for the more experienced Apple user. Each chapter proceeds in detail through each application package, describes a tutorial (on diskette), presenting ways to customize each package to reflect individual needs, and discussing possible problems and their solutions. One chapter describes a set of utility programs provided with the system. These programs permit you to duplicate disks, make backup copies, determine how many entries have been made, and so on. An appendix lists all of the reports available within each of the programs and provides a detailed sample of each report. These samples are clearly labeled and explained. Another appendix provides a copy of all of the menus—something you're likely to find particularly useful. Finally, a list of frequently asked questions, along with their answers, is included.

The package comes complete with program diskettes, data diskettes, tutorial diskettes, and backup diskettes. Diskette envelopes (three-ring binder type) and labels are also provided. A set of continuous business forms is included with your original package so that you can begin to use it immediately.

One way to make a quick evaluation of this sort of package is to examine the reports it can produce. *The Controller* can provide a business with the capacity to generate these reports:

- general ledger
- accounts payable
- chart of accounts with changes
- chart of accounts with balances
- standard entries journal
- general journal report
- detail trial balance
- detail balance sheet
- customer mailing labels

vendor mailing labels
open invoices by vendor
aged trial balance
detail income statement
income statement for departments
detail income statement for departments
detail aged trial balance
new A/P invoice register
A/P cash requirements
A/P paid invoice register
A/P distribution journal
A/P disbursement journal
accounts receivable
chart of accounts
trial balance
balance sheet
income statement
sales journal
payment journal
customer list
vendor list
checks & stubs
sales commissions
monthly activity

The Readers Speak. "If I wanted to put together an Apple system for use in a business, what would you describe as the 'ideal' system?" J.K. Chicago, IL

There are perhaps as many answers to that question as there are businesses in which the Apple can be used. However, it's possible to describe what might be considered an effective system. This description must be tempered, of course, by the unique characteristics of the business. What volumes of data are involved? Is this an in-home business or a $20 million corporation? In any case, here are some suggestions.

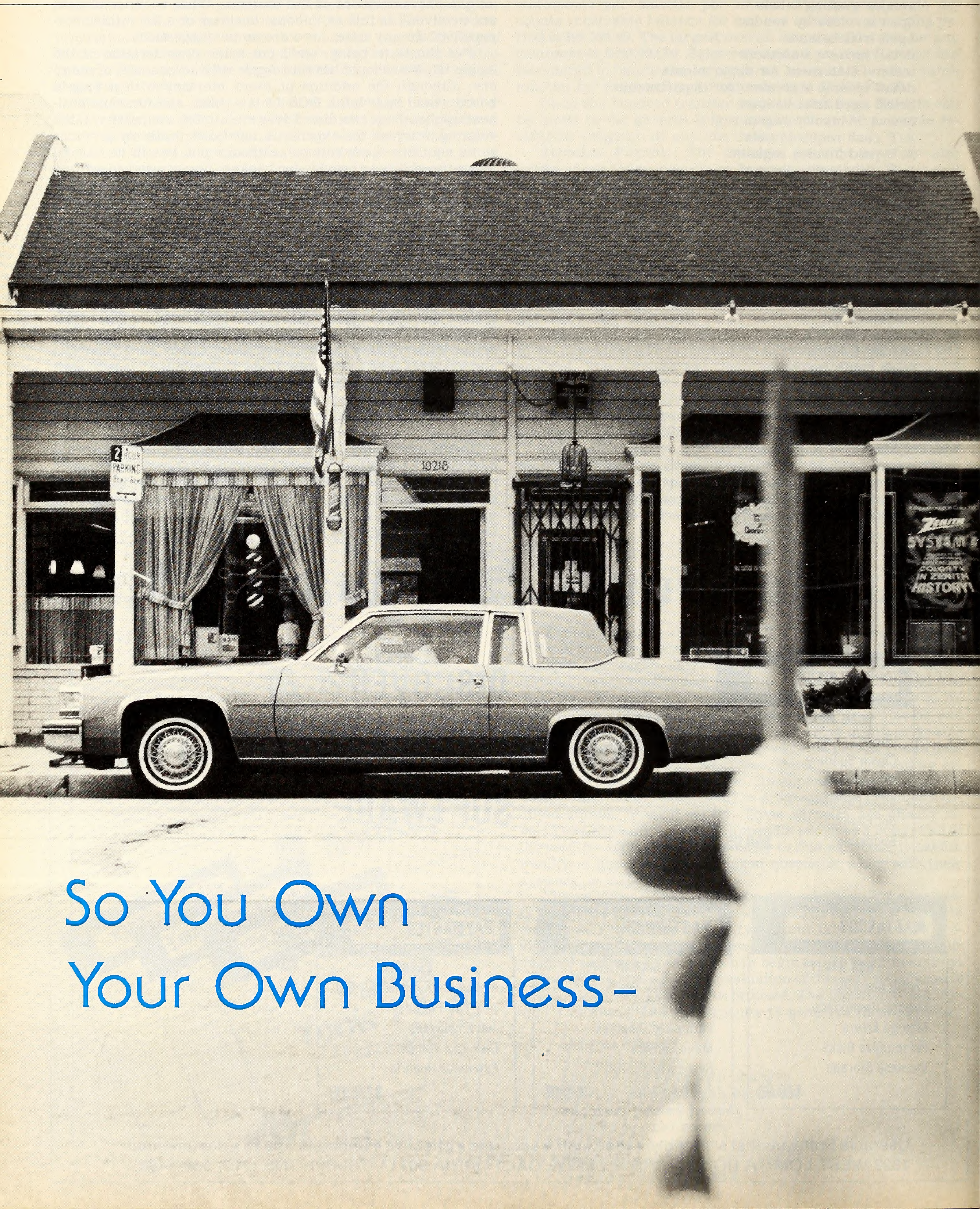
For the time being, we'll put aside consideration of the Apple III. You should have an Apple with at least 48K of memory, although the addition of more memory with a plug-in board would be helpful. DOS 3.3 is a must, and for most business applications, two disk drives are almost a necessity. If the volumes warrant the expense, a hard disk certainly provides some significant advantages, although one has to be careful about the ability to provide for backup. For general reporting, word processing, and record keeping, a printer is a must. If high quality is desired, then you must obtain a letter-quality printer.

Some additional items worth considering: a communications card (for connecting to a modem and accessing a computer service or another available computer), a graphics tablet, and a built-in clock.

Here are two business systems. The first would be an ideal system and the second a more practical alternative. The ideal: An Apple II Plus with 48K, twenty million-character hard disk drive, letter quality printer, communications card, modem, and graphics tablet. The practical: An Apple II Plus with 48K, two floppy disk drives, and a draft printer.

Rumor has it that Apple will soon be unveiling some surprises that may significantly alter the suggestions offered here.

In the meantime, have a nice holiday. Next month we'll evaluate *The Controller* package and Westware Systems II package. In addition, we'll discuss the price-versus-service decision a person is faced with when buying a computer system.



So You Own Your Own Business –

Like many things owned by people, a business is a difficult possession to put a value on. Yet the dollar value of a business is important to know. It may be needed for determining a price for a buy/sell agreement, for life insurance purposes, or for tax reasons.

Your Apple can provide a useful means for evaluating your business through a simple program. The program determines an average business value based on four formulas frequently used by tax authorities and businesspeople. The methods are equally valid for a sole proprietorship, a partnership, or a corporation, provided the input properly reflects the type of business involved. The Applesoft program requires an eighty-column printer or wider. If the print head is set at the top of the page, the output will easily fit an eight-by-eleven-inch page.

The program is intended to be a turnkey program calling for interaction between you and your computer. Immediately after the last item of input is completed, the output will be sent to the printer. It should be noted that the print command—PR#(slot)—is the command for a parallel printer. Appropriate commands for a serial printer should be added to the program if necessary.

Input Inquisition. The input items requesting the name of the business and the slot number for the printer card are self-explanatory. *Book Value*, another input item, is easily obtained from the company's balance sheet by subtracting total liabilities from total assets. The item called *overvalued assets* allows you to reflect assets overvalued in the balance sheet such as obsolete business furniture or equipment (such as pre-Apple computers). Similarly, *undervalued assets* reflects assets depreciated for accounting purposes that are of greater actual value. A building is often such an asset.

Average annual earnings depends on the kind of business involved. If it's a corporation, after-tax earnings should be used; if it's a sole proprietorship or partnership, pretax earnings should be used. The average annual earnings should include at least five years if possible, although you may wish to weight some years over others.

The item requesting excess owner compensation asks you whether outside managers could do the same job for less pay. Put more bluntly, is the owner pulling out more in salary than would be paid an outside salaried employee? Conversely, *additional compensation required for outside management* asks if outsiders taking over the management duties would require a higher salary than the owner's current salary. In many cases, especially with large or publicly held companies, both these input items would be *0*. Where a sole proprietorship or partnership is involved, salary should be equated with the owner's annual draw on the business (unless salary schedules exist).

WHAT'S IT WORTH?

BY STEPHEN PARRISH

Percent fair return on secure investment asks you what a secure medium to long-term return on investment would be. It's asking what reasonable return would be obtained if the company were liquidated and the proceeds invested. To provide an accurate valuation, this percentage rate should normally be relatively low. In contrast, *percent fair return on investment similar to this business* asks what an investor would demand and get for an investment similar to the business involved. Thus, the percentage rate would normally be higher than the earlier percentage rate because most businesses involve more risk than a secure long-term investment.

Good Will Is Worth Money. *Good will stability factor* is, perhaps, the most difficult input item. Good will is the intangible extra value company managers bring to a business. Although the program calculates the monetary value of this good will, it remains to be determined how long this good will would last if the managers were gone. In other words, how long would the excess value attributable to, say, Smith and Sons last if Smith and Sons were themselves gone? A generally accepted means of ascertaining this number is to tie it to the stability of the business. For example, a medical supply company in Rochester, Minnesota, home of the Mayo Clinic, would probably have a stable future. An appropriate stability factor would be *7*. In contrast, a one-person law firm would lose most, if not all, of its good will upon the death of its owner. Since its stability after the owner leaves it is minimal, a *3* would be an appropriate good will stability factor (if you were valuing for buy/sell purposes). Small but stable businesses should use a midrange entry such as *5*.

Investors seeking to invest in an ongoing business are generally most interested in the earnings the business can generate. Thus, the investor is, in a way, seeking to buy part of the company's future earnings—that part of earnings that exceeds what the investor can obtain from a secure investment (good will).

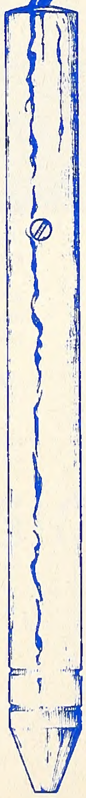
Years earnings to be purchased asks how many years of excess earnings the investor would demand. The appropriate number of years to enter depends on the nature of the business, and the business owner would normally have a good feel for what this number should be. Because the years purchased method is inapplicable to some businesses, a *0* entry will eliminate this method from the valuation formula.

```
                  VALUATION MODEL OF
                   WOODWORTH INC.
**************************************************************
ADJUSTED BOOK VALUE *
**********************
  BOOK VALUE                          1200000
  UNDERVALUED ASSETS                + 30500
  OVERVALUED ASSETS                 - 10000
                                    $1220500
**************************************************************
CAPITALIZED GOODWILL PLUS BOOK VALUE *
****************************************
  AVRG. ANN. EARNINGS                 125300
  EXCESS OWNER SALARY               + 40000
  ADDTL REPLACEMENT SALARY          - 0
  ADJUSTED EARNINGS                 = 165300
  FAIR RTN. ON BOOK AT 8%           - 95999
  AVRG. ANN. GOODWILL               = 69301
  GOODWILL MULTIPLIER               * 5
  CAPITALIZED GOODWILL              = 346505
  ADJUSTED BOOK VALUE               + 1220500
                                      $1567005
**************************************************************
CAPITALIZED EARNINGS *
*********************:
  ADJUSTED EARNINGS                   16530
  CAPITALIZATION RATE
  1/12%—FAIR RTN. ON RISK            * 8
                                    $1084166
**************************************************************
TIMES YEARS PURCHASE *
**********************
  AVRG. EXCESS RETURN OVER SECURE INVESTMENT
    SEE GOODWILL ABOVE                69301
  YEARS BEING PURCHASED             * 3
                                    $207903

                 AVERAGE OF ABOVE VALUES
                        $1019893
```

It's important to realize that some of the input items are not precisely measurable. However, the combining and averaging of these data can provide a useful means of putting a relatively accurate dollar value on the business.

Jumping to Conclusions: Output. *Adjusted book value* is the most basic measure of a business value. It simply takes assets and liabilities from the balance sheet and adjusts the values to reflect true values rather than accounting values. It's the current market value of assets less any liabilities. Although real values are used, it does not attribute any value to good will or to the earnings history of the business. Other than for valuing a liquidating business, its primary usefulness is as a component of other valuation methods.

Capitalized good will plus book value assumes there are two components making up the total value of a business: adjusted book value plus the capitalized value of good will. To arrive at the value of good will, the earnings of the business are separated into two components: earnings that represent a reasonable return on book value based on returns for secure money investments, and earnings that are due to good will. Earnings in excess of a reasonable return on book value are attributed to



LETTER PERFECT

T.M. LJK

WORD PROCESSING

ATARI 400/800 — APPLE II & II+

EASY TO USE — Letter Perfect is a single load easy to use program. It is a menu driven, character orientated processor with the user in mind. FAST machine language operation, ability to send control codes within the body of the program, mnemonics that make sense, and a full printed page of buffer space for text editing are but a few features. Screen Format allows you to preview printed text. Indented margins are allowed. Data Base Merge with **DATA PERFECT** by LJK, form letters, accounting files and mailing labels only with MAIL MERGE/UTILITY by LJK. **FEATURES** — Proportional/Incremental spacing * Right Justification * File Merging * Block movement * Headers * Footers * Print Multiple Copies * Auto Page Numbering * Scroll forward/backward * Search and Replaces * Full cursor control * Underlining * Boldface * Superscripts * Subscripts * Auto page numbering * Insert character/line * Delete character/line * Centering * Horizontal tabs/changeable * Multifunction format line (line spacing — left margin — page width — lines/page — change fonts — top/bot margin adjust) **MUCH MORE! $149.95**

ATARI VERSION 2.0 #2001

Compatible with Atari DOS. Uses proportional font, right justified with Atari 825/Centronics 737, 739 printers. Uses EPSON MX* Series + Graftrax/italicized font. Can mix type fonts on same page; mix boldface and enhanced font in same line with justification. Can be used with 16K Atari/400.*

"Compared to the price of many other word processors, this package is a steal. It does everything the advertisement claims and more. On top of this the software is very easy to use." **A.N.A.L.O.G. MAGAZINE**

APPLE VERSION 5.0 #1001

DOS 3.3 compatible — Use 40 or 80 column interchangeably (Smarterm — ALS; Videoterm-Videx; Full View 80 — Bit 3 Inc.; Vision 80 — Vista; Sup-R-Term — M&R Ent.) Reconfigurable at any time for different video, printer, or interface. **USE HAYES MICROMODEM II*** LCA necessary if no 80 column board, need at least 24 K of memory. Files saved as either Text or Binary. Shift key modification allowed. Data Base Merge compatible with **DATA PERFECT*** by LJK.

"For $150, Letter Perfect offers the type of software that can provide quality word processing on inexpensive micro-computer systems at a competitive price." **INFOWORLD**

DATA PERFECT T.M. LJK

APPLE & ATARI DATA BASE MANAGEMENT — $99.95

Complete Data Base System. User orientated for easy and fast operation. 100% Assembly language. Easy to use. You may create your own screen mask for your needs. Searches and Sorts allowed, Configurable to use with any of the 80 column boards of Letter Perfect word processing, or use 40 column Apple video. Lower case supported in 40 column video. Utility enables user to convert standard files to Data Perfect format. Complete report generation capability. **Much More!**

EDIT 6502 T.M. LJK

This is a coresident — two pass **ASSEMBLER, DIS-ASSEMBLER, TEXT EDITOR, and MACHINE LANGUAGE MONITOR.** Editing is both character and line oriented. Disassemblies create editable source files with ability to use predefined labels. Complete control with 41 commands, 5 disassembly modes, 24 monitor commands including step, trace, and read/write disk. Twenty pseudo opcodes, allows linked assemblies, software stacking (single and multiple page) plus complete printer control, i.e. paganation, titles and tab setting. User can move source, object and symbol table anywhere in memory. Feel as if you never left the environment of **BASIC.** Use any of the 80 column boards as supported by **LETTER PERFECT.** Lower Case optional with **LCG.**

LJK DISK UTILITY — APPLE $29.95

This menu driven program allows the user to manipulate a variety of different file types. Binary, Text, and Source files may be easily converted into each other. The program may be used with **APPLESOFT*, VISCALC*,** and other programs. These program files may be readily adapted for multiple use including editing with **LETTER PERFECT** word processings.

MAIL MERGE/UTILITY — APPLE & ATARI — $29.95

This menu driven program combined with **LETTER PERFECT** allows user to generate form letters and print mailing labels. With the Atari, you may **CONVERT ATARI DOS FILES,** or Visicalc files compatible for editing with **LETTER PERFECT.** Utility creates Data Base files for Letter Perfect.

LOWER CASE CHARACTER GENERATOR — $34.95

```
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFG
HIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmno
pqrstuvwxyz{|}~
```

Lower Case Character Generator for the Rev. 7, Apple II or II+ computers. When installed, this Eprom will generate lower case characters to the video screen. Lower case characters set has two dot true descenders. Installation instruction included. Manual includes listing of software for full support and complete instructions for shift key modification. Compatible with **LETTER PERFECT.**

COMPUTER BASED SOFTWARE — LJK™ ENTERPRISES INC.

LJK ENTERPRISES INC.
P.O. Box 10827
St. Louis, MO 63129
(314) 846-6124

DEALER INQUIRES INVITED

*Trademarks of: Apple Computer — Atari Computer — Epson America — Hayes Microcomputers — Personal Software — Videx — Bit 3 Inc. — M&R Ent. — Advanced Logic Systems — Vista Computers

YOUR APPLE CAN RUN UP TO TWENTY TIMES FASTER FOR ONLY $99.95

Expediter II provides an easy means by which Applesoft BASIC programs written for the Apple II & III* Computer may be translated into machine language. As a result, the compiled version of the program will normally execute from 2-20 times faster. All features of Applesoft are fully supporting including the use of the Hi-Res graphics, shape, tables, Lo-Res graphics, defined functions and DOS commands.

There is no additional BASIC syntax to learn, and most compilations can be performed with one simple command.

FEATURES . . .

- Debugging of compiled code is aided through use of such tools as TRACE capabilities and a complete symbol table listing.
- All DOS functions are supported in exactly the same way as in the uncompiled BASIC program-nothing new to learn.
- All standard Applesoft error messages are supported.
- Erroneous statements are flagged during program compilation.
- Extensive use is made of pre-existing routines within the Applesoft ROM. This minimizes the memory required by compiled machine code.
- Comprehensive statistics are generated at compile times such as total memory requirement, starting address of the compiled machine code for each line of the Applesoft program, etc.
- Applesoft programs may be compiled to reside at any location in the Apple's memory. This feature permits multiple Applesoft programs to overlay each other and use the compiled Applesoft programs as subroutines.
- Variables may be designated as "local" to an Applesoft program or "global" to all Applesoft programs in memory, thereby facilitating inter-program communications.
- Compiles BASIC programs at approximately 30 lines per minute.

EXPEDITER II

The Applesoft Compiler by* ON-LINE systems

EXPEDITER II is a registered trade mark of Einstein/Goodrow

Expediter II runs on any Apple computer with Applesoft in ROM, 48K and a disk drive. It is available NOW for the introductory price of $99.95. * Apple II, Apple III and Applesoft Basic are registered trade marks of Apple Computers.

ON-LINE SYSTEMS · 36575 Mudge Ranch Road · Coarsegold, CA 93614 · 209-683-6858

Visa, Master Charge, COD or Checks Accepted

earnings on good will. These earnings are capitalized by multiplying them by the number of years that good will is reasonably expected to last. The capitalized value of good will is then added to the adjusted book value to arrive at the total value of the business.

Capitalization of earnings ignores book value and values the business solely by capitalizing its earnings. The capitalization rate is the reciprocal of the expected rate of return on invested capital in businesses of a similar nature. Thus, this method calculates the amount of capital required to yield a return equal to average annual earnings.

Years earnings purchased is similar to the capitalized good will method. It calculates the annual earnings in excess of a secure return on capital. These excess earnings are then multiplied times the number of years to be purchased. The total value of the business is derived by adding the number of years excess earnings to be purchased and the adjusted book value.

It's Time To Type.

```
10    HOME : FOR X = 1 TO 10: PRINT : NEXT X
20    PRINT " A BUSINESS VALUATION PROGRAM BY"
30    PRINT : PRINT " STEPHEN B. PARRISH, J.D.,C.L.U."
40    FOR X = 1 TO 3000: NEXT X: HOME
105   INPUT "NAME OF BUSINESS? ";BUSNESS$
106   PRINT : INPUT "WHAT SLOT IS USED FOR THE PRINTER? ";SLOT
109   PRINT : PRINT : PRINT
110   INPUT "BOOK VALUE (ASSETS LESS LIABILITIES ON BALANCE SHEET)?
        ";BOOK
120   PRINT : INPUT "AMOUNT ASSETS ARE OVERVALUED? ";OVERVL
130   PRINT : INPUT "AMOUNT ASSETS ARE UNDERVALUED? ";UNDERVL
140   PRINT : INPUT "AVERAGE ANNUAL EARNINGS? ";EARNINGS
150   PRINT : INPUT "EXCESS OWNER COMPENSATION? ";EXCESSCOMP
160   PRINT : INPUT "ADDITIONAL COMPENSATION REQUIRED FOR OUTSIDE
        MANAGEMENT? ";ADDTLCOMP
165   PRINT : INPUT "PERCENT FAIR RETURN ON A SECURE INVESTMENT?
        ";SECURE
170   PRINT : INPUT "PERCENT FAIR RETURN ON INVESTMENT SIMILAR TO THIS
        BUSINESS? ";RTN
175   PRINT : INPUT "GOODWILL STABILITY FACTOR (SPECULATIVE=3; HIGHLY
        STABLE=7)? ";STABILITY
180   PRINT : INPUT "YEARS EARNINGS TO BE PURCHASED (TYPE 'O' IF
        INAPPLICABLE)? ";YRS
3000  PR# (SLOT)
3005  SPEED= 200
3010  FOR W = 1 TO 5: PRINT : NEXT W
3020  PRINT TAB ( 31) "VALUATION MODEL OF"
3030  PRINT TAB( 40 - (.5 * ( LEN (BUSNESS$) ) ) )BUSNESS$
3040  PRINT : PRINT : PRINT "******************************************
        ************************"
3050  PRINT "ADJUSTED BOOK VALUE *"
3060  PRINT "*********************"
3070  TAB( 4)"BOOK VALUE",," ";BOOK
3080  PRINT TAB( 4)"UNDERVALUED ASSETS","+ ";UNDERVL
3090  PRINT TAB( 4)"OVERVALUED ASSETS","- ";OVERVL
3095 ABV = BOOK + UNDERVL - OVERVL
3100  PRINT ,,"$";BOOK + UNDERVL - OVERVL
3110  PRINT " ****************************************************
        ****************************"
3115  PRINT "CAPITALIZED GOODWILL PLUS BOOK VALUE *"
3120  PRINT "**************************************"
3130  PRINT TAB( 4),, "$";FF + (BOOK + UNDERVL - OVERL)
3140  PRINT TAB( 4)"EXCESS OWNER SALARY","+ ";EXCESSCOMP
3150  PRINT TAB( 4)"ADDTL REPLACEMENT SALARY","- ";ADDTLCOMP
3155 AE = EARNINGS - ADDTLCOMP + EXCESSCOMP
3160  PRINT TAB( 4)"ADJUSTED EARNINGS",,"= ";AE
3170  PRINT TAB( 4)"FAIR RTN. ON BOOK AT ";SECURE;"%","- "; INT (BOOK *
        (SECURE / 100) )
3180  PRINT TAB( 4)"AVRG. ANN. GOODWILL","= ";AE - ( INT (BOOK *
        (SECURE / 100) ) )
3190  PRINT TAB( 4)"GOODWILL MULTIPLIER","* ";STABILITY
3195 FF = STABILITY * (AE - ( INT (BOOK * (SECURE / 100) ) ) )
3200  PRINT TAB( 4)"CAPITALIZED GOODWILL",,"= ";FF
3210  PRINT TAB( 4)"ADJUSTED BOOK VALUE","+ ";BOOK + UNDERVL -
      OVERVL
3215 CGPBV = ABV + FF
3220  PRINT TAB( 4),, "$";FF + (BOOK + UNDERVL - OVERVL)
3225  PRINT
3230  PRINT " ****************************************************
        *************************"
3240  PRINT "CAPITALIZED EARNINGS *"
3250  PRINT "**********************"
3260  PRINT TAB( 4)"ADJUSTED EARNINGS"," ";EARNINGS + EXCESSCOMP -
        ADDTLCOMP
3270  PRINT TAB( 4)"CAPITALIZATION RATE": PRINT TAB( 4)"1/";RTN;"% - FAIR
        RTN. ON RISK"," * "; INT (100 / RTN)
3275 CE = INT ( (100 / RTN) * EARNINGS + EXCESSCOMP - ADDTLCOMP)
3280  PRINT ,,"$";CE
3290  PRINT " ****************************************************
        **************************"
3295  IF YRS = 0 GOTO 3360
3300  PRINT "TIMES YEARS PURCHASE *"
3310  PRINT "**********************"
3320  PRINT TAB( 4)"AVRG. EXCESS RETURN OVER SECURE INVESTMENT"
3330  PRINT TAB( 6)"SEE GOODWILL ABOVE"," ";FF / STABILITY
3340  PRINT TAB( 4)"YEARS BEING PURCHASED","* ";YRS
3350 YP = INT (YRS * (FF / STABILITY) )
3355  PRINT ,,"$";YP
3360  PRINT : PRINT : PRINT : PRINT
3370  PRINT TAB( 15)"=================================
        ====================="
3380  PRINT TAB( 31)"AVERAGE OF ABOVE VALUES"
3382  IF YRS = 0 GOTO 3395
3385  PRINT
3390  PRINT TAB( 36)"$"; INT ( (ABV + CGPBV + CE + YP) / 4): GOTO 3412
3395  PRINT TAB( 36)"$"; INT ( (ABV + CGPBV + CE) / 3)
3412  FOR X = 1 TO 10: PRINT : NEXT X
3415  PRINT "-"
3416  SPEED= 255
3417  PR# 0
3420  HOME : INVERSE : PRINT : PRINT : PRINT : PRINT :PRINT " AGAIN? '1' IF YES
        '2' IF NO "
3430  INPUT UU
3440  IF UU = 1 THEN 4000
3450  PRINT "DO YOU WISH TO RUN A NEW VALUATION? '1' IF YES '2' IF NO
3460  INPUT YY
3470  IF YY = 1 GOTO 5000
3480  HOME : PRINT : PRINT : PRINT : PRINT : PRINT "BYE!"
3485  NORMAL
3490  END
4000  NORMAL : HOME : GOTO 3000
5000  NORMAL : HOME : GOTO 105
```

THE BASIC Solution

By Wm. V. R. Smith

The Basic Solution has received many interesting letters from *Softalk* readers. Many of you have suggested methods to solve some of the programming problems discussed in previous Basic Solutions. Each response will be reviewed, and the best routines will be reprinted in future articles.

In this month's Basic Solution, we'll discuss the brute force method of programming. Most new programmers end up creating programs using the brute force method.

An anecdote—a true story, incidentally—will help define the brute force method of programming:

There was once a teacher who was instructing young students in the principles of mathematics. One day, he wanted to take a break from class and go for a cup of coffee. He decided to assign the students a problem to occupy them while he slipped away to enjoy a steaming brew. The problem he chose to assign was to add all the numbers from 1 through 100. This teacher was relying on the assumption that it would take the students quite a while to add together a hundred numbers—1 plus 2 plus 3 plus 4, and so on.

The laborious task of adding the numbers together one at a time is a perfect example of brute force.

In this class, however, was a child prodigy who found a quicker method to solve the problem. He wrote down on his answer sheet the following formula: 1 plus 99, 2 plus 98, 3 plus 97, 4 plus 96 . . . 49 plus 51.

Thus, he saw that ninety-eight of the numbers to be added could be paired to equal 100 per pair. Forty-nine pairs equalled 4,900. He then added the 50 and the 100 and came up with 5,050. He marched up to the teacher, who was about to leave the room, and handed in his result. Needless to say, the coffee break was over before it started.

The first reader from each continental United States time zone to send in the name of this young student to the Basic Solution will receive a $10 credit toward any purchase at his/her local computer store. (Alaska, Hawaii, and other countries will be counted with Eastern Time Zone entries.)

In this instance, the brute force method would have taken quite a bit of time to complete. To continue the concept of brute force to computer programming, the Basic Solution submits the following problem:

Most chess players are familiar with the knight's tour, in which a knight is placed in its proper position on an empty chessboard and then is moved in the proper knight fashion to every square on the board without repeating.

This month's Basic Solution draws a chessboard on the Apple screen, places the knight in its proper position, and attempts every possible move until the board is filled. If the knight can no longer move, and there are spaces still left empty, the knight will retrace his moves and try every other possible direction. This will continue until even the first move is subtracted, and a different direction is used. This method of solving the knight's tour is another example of brute force. The code is fairly short and should be easy to type in. The amount of time it takes to complete the tour is unknown.

Southwestern Data System's *Speed Star* Applesoft compiler was used on the knight's tour with a significant increase in execution speed. If you have access to any of the new compiler programs, the knight's tour will provide you with a nice timing gauge.

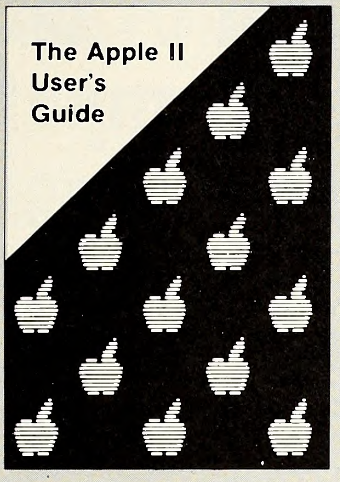
Lines 330 and 340 may be modified to experiment with a smaller chessboard. Line 205 controls the starting position of the knight on the board.

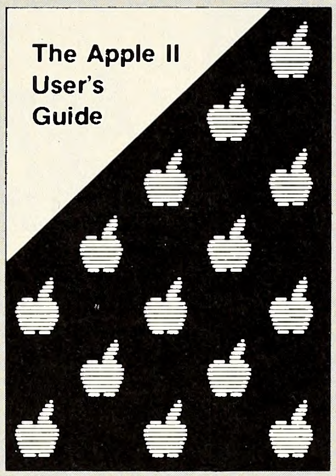
If you can find a faster method than brute force for solving this particular problem, please send it with your name and address for possible use in a future Basic Solutions.

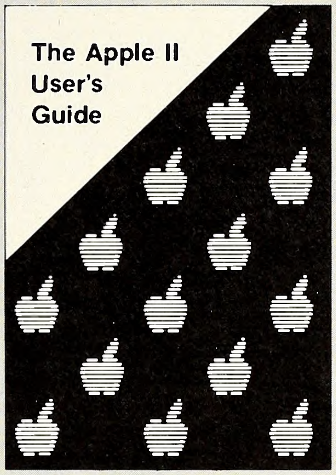
Your comments and solutions to problems are always welcome. If your solution is printed you will receive a $10 credit toward any purchase at your local computer store. Mail your input to *Softalk* Basic Solution, 11021 Magnolia Boulevard, North Hollywood, CA 91601.

```
10   DIM A(64,3),C(8,8)
100  FOR X = 1 TO 64
110  FOR Y = 1 TO 3
120 A(X,Y) = 0
130  NEXT Y
140  NEXT X
150  DATA 1,−2,2,−1,2,1,1,2,−1,2,−2,1,−2,−1,−
       1,−2
160  FOR X = 1 TO 8
170  READ B(X,1),B(X,2)
180  NEXT X
190  HOME
195  GOSUB 600
200 P = 1
205 A(0,1) = 7:A(0,2) = 10
210 A(P,3) = A(P,3) + 1
215  HTAB 1: VTAB 20: PRINT "WORKING ON
        NUMBER ";P
220  IF A(P,3) < 9 THEN 300
230  FOR X = 1 TO 3
240 A(P,X) = 0
250  NEXT X
260 P = P − 1
263  VTAB A(P,2) * 2: HTAB A(P,1) * 3
264  PRINT ". ";
265 C(A(P,1),A(P,2)) = 0
270  IF P > 0 THEN 210
280  PRINT "P=0 AT LINE 280": END
300  REM
310 A(P,1) = A(P − 1,1) + B(A(P,3),1)
320 A(P,2) = A(P − 1,2) + B(A(P,3),2)
330  IF A(P,2) > 8 THEN 500
340  IF A(P,1) > 8 THEN 500
350  IF A(P,1) < 1 THEN 500
360  IF A(P,2) < 1 THEN 500
370  IF C(A(P,1),A(P,2)) < > 0 THEN 500
380 C(A(P,1),A(P,2)) = P
400  IF P = 64 THEN 900
410 X1 = A(P,1) * 3:Y1 = A(P,2) * 2
420  VTAB Y1: HTAB X1: PRINT P;
440 P = P + 1
450  GOTO 210
500  GOTO 210
600  FOR X = 1 TO 8
610  FOR Y = 1 TO 8
620  VTAB Y * 2: HTAB X * 3
630  PRINT ".";
640  NEXT Y
650  NEXT X
660  RETURN
```

Osborne Polishes THE APPLE







and Publishes THE GUIDE

The Apple II User's Guide by Lon Poole, Martin McNiff, and Steven Cook #46-2, $15 ☐

This Guide is the key to unlocking the full power of your Apple II or Apple II plus computer. The Apple II User's Guide brings together in one place a wealth of information for Apple computer users. It will tell you more about your Apple than any other single source.

Topics include: **Applesoft and Integer BASIC programming** - especially how to make the best use of Apple's sound, color and graphics capabilities. The book presents a thorough description of every BASIC statement, command and function. **Machine level programming** - although not a machine language programming guide, this book covers the Machine Language Monitor in detail. **Hardware features** - the disk drive and printer are covered in separate chapters. **Advanced programming** - special sections describe high resolution graphics techniques and other advanced applications.

This book will save you both time and effort. No longer will you have to search endlessly for useful information. It's all here, in the Apple II User's Guide, thoughtfully organized and easy to use.

*Apple is a trademark of the Apple Computer Corporation.

More Osborne Books for Apple Users

6502 Assembly Language Programming
by Lance Leventhal

Here's the book that shows you how to program the Apple in assembly language. "With its numerous reference tables, explanations and descriptions, this book is a major accomplishment. It's by far the most complete 6502 book to appear. Make space on your bookshelf. You'll want to keep this one handy." *COMPUTE*

Book #27-6 $16.99, 606 pp ☐

Some Common BASIC Programs
by L. Poole and M. Borchers

Written in a subset of standard BASIC that is compatible with many microcomputers, this book contains 76 useful and easy-to-use programs. The book explains how each program works, shows sample runs, and has program listings with remarks. "If you want to do mathematics or statistical programming on your computer, this book is a must." *ON COMPUTING* Magazine

Book #06-3 $14.99 200 pp ☐

Practical Basic Programs
edited by Lon Poole

Here's a collection of 40 programs you can easily key in and use on most microcomputers. *Practical Basic Programs* is especially useful in small business applications. It solves problems in finance, management, and statistics. The book contains sample runs, practical problems, BASIC source listings, and an easy-to-follow narrative to help you realize the potential uses of each program.

Book #38-1 $15.99 200 pp ☐

An Introduction to Microcomputers Volume I - Basic Concepts Second Edition
by Adam Osborne.

The world's best selling textbook on microcomputers develops a detailed picture of what a microcomputer can do, how it does what it does, and how it can be used in any practical environment. This is the most comprehensive and up-to-date introduction to microprocessor systems available anywhere.

Book #34-9, $12.99, 320 pp ☐

Make check payable to: OSBORNE/McGraw-Hill
630 Bancroft Way, Berkeley, CA 94710 **Dept. AS4**

Phone Orders: (415) 548-2805 or Toll-Free 800-227-2895

Name ______________________________

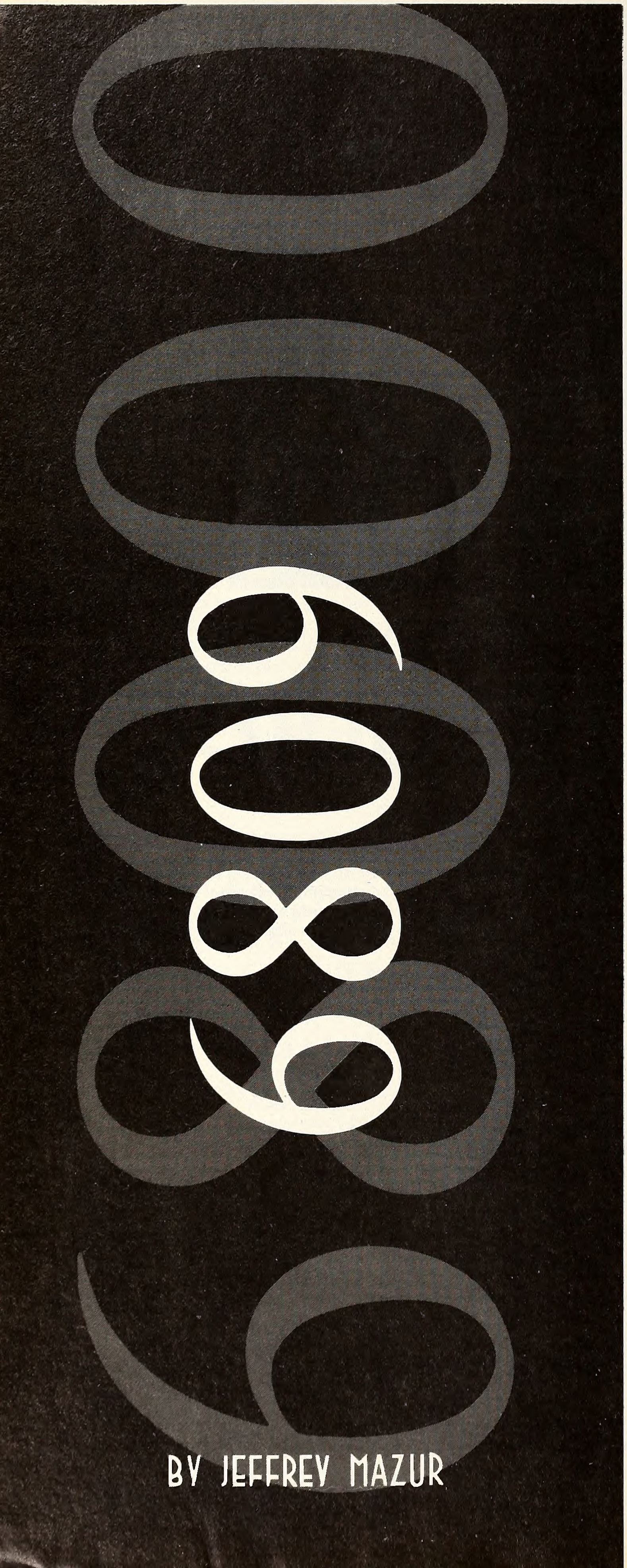
Address ______________________________

City/State/Zip ______________________________

Plus ☐ .75/item 4th class ☐ $1.50/item UPS ☐ $2.50/item Air Mail ☐ $10.00/item Overseas (California residents add applicable tax.)

Total Amount Enclosed $ __________ or charge my ☐ VISA ☐ Mastercharge Card # __________ Exp. Date __________

Signature ______________________________ ☐ Please send me your free catalog.



BY JEFFREY MAZUR

GRIST FOR THE MILL

And Other Apple-Compatible Microprocessors

When Stellation Two announced the Mill some months ago, it did so with little fanfare and no software. It wasn't that the coprocessor designed to give the Apple II a coresident 6809E CPU didn't deserve fanfare, but without software, the Mill was like a powerful engine without an automobile to run.

Now the automobiles are beginning to arrive. Two important software packages have emerged to run with the Mill, *The Pascal Speed-Up Kit* and an assembler development kit.

At the same time, recent history is beginning to repeat itself. Another attached-processor board for the Apple has been announced, and the significance of this one calls for fanfare regardless of software packages. This one is the 68000 sixteen-bit (pseudo thirty-two-bit) CPU from Digital Acoustics, alias Dtack Grounded. Its powers far outreach those of mortal 6502s and even 6809s.

Faster Pascal with the Mill. The *Pascal Speed-Up Kit* from Stellation Two and SB Programming consists of a single disk and one page of instructions. Setting up Pascal to use the Mill is accomplished by executing the *Install* program supplied on the disk. This makes changes in the file SYSTEM.APPLE to allow operation with the 6809. You will be prompted to select the slot in which the Mill will be placed and whether or not you would like a running indication of which CPU is in control. The entire modification process takes about three minutes.

One of the features Pascal is most noted for is its portability—that is, the ability to write a program on one type of computer and have it run on another without modification. This is accomplished by having the original program (source code) compiled into intermediate machine language-like p-code. This p-code is not directly executable on most computers (with the exception of a few true p-machines like Western Digital's Pascal Engine). Instead, the p-code is translated into the computer's native language by a p-code interpreter. This is similar to the process by which the Integer and Applesoft interpreters convert Basic programs to 6502 machine language. Since the p-code for a given program does not change, it can be run on any computer assuming that the appropriate p-code interpreter exists.

What the *Speed-Up Kit* does basically is replace the 6502 p-code interpreter in the file System.Apple with a 6809 interpreter that runs on the Mill. Because the 6809 runs more efficiently than the 6502 on the average, an overall speed improvement can be realized. The company claims that the 6809 will run between 30-300% faster. However, due to I/O and other 6502 operations plus the overhead of supporting the 6809, total improvement will be considerably less. Since the *Speed-up Kit* only modifies the p-code interpreter, all the operating system, utility, and program code files will run faster without any modification. (You did know that the Pascal Operating System, Filer, Editor, and all the rest are written in Pascal and therefore execute via the p-code interpreter, didn't you?)

The beauty of the *Speed-Up Kit* is its simplicity. Once the modified System.Apple file is created, everything runs nor-

EMPIRE I: WORLD BUILDERS™

By David Mullich, author of THE PRISONER

WORLD BUILDERS, the colonial stage of the Empire history... the first Interactive Fantasy of the EMPIRE GAMING TRILOGY. Join the travelers, scouts, builders, plunderers, asteroid miners, and high-technology homesteaders... pioneers on the fringes of the galaxy... citizens of an Empire still in its infancy. You can be any of them... you can be all of them. Your goals are to survive, to conquer, to prosper.

Explore the Empire using WORLD BUILDER's extensive English vocabulary. Pilot your All Terrain Vehicle (ATV) through treacherous swamps or rugged mountains, or build a thriving community in the midst of a desert... but beware...

You will traverse weird and fantastic landscapes draped in spectacular colors, thanks to a new, sophisticated high-resolution graphics system developed by the makers of IFs. And more: a soft-cover "Chronicle" retraces the EMPIRES's history from its inception to the time of the WORLD BUILDERS.

The EMPIRE GAMING TRILOGY will be completed by INTERSTELLAR SHARKS™ and ARMAGEDDON.™

Applesoft, 48k, DOS 3.3.....$32.95 — if™ INTERACTIVE FANTASIES™ — Available at your local computer store.

Edu-Ware Services, Inc. 22222 Sherman Way Suite 102 Canoga Park, CA 91303 (213) 346-6783

mally, except a little faster. Although the actual speed improvements realized in our tests were sometimes disappointing, any improvement would be worthwhile if you already have the Mill in your Apple. Equally impressive is the ease with which you can revert to normal Mill-less operation.

If you ever need to remove the Mill or just don't want to use it, simply swap back the old System.Apple file.

The following data represents the actual performance of the *Speed-Up Kit* on several programs. Of course, the total speed improvement depends on the particular program or task, but our overall opinion of the kit is that it will speed things up from 10 to 15 percent. Since this is accomplished with both hardware and software, it is conceivable that further improvements may be achieved in later revisions of Pascal and/or the *Speed-Up Kit*.

The first test was to determine the speed of the compiler running with the Mill. A very large program was used to make the comparison less critical. This program, called *Prose*, is a public domain text editor which contained a total of 3,968 lines of Pascal source code. Normally this program takes thirteen minutes, forty-two seconds to compile; with the speed-up, it took only eleven minutes, fifty-three seconds. This amounts to a 13 percent speed-up which should be a fairly good value for system functions.

Next, several of the demonstration programs on the Apple3: disk were tried. The greatest speed improvement was found in the GrafDemo program where the image of a butterfly is created. Normally this takes 22.2 seconds (all data manipulation—no I/O involved). With the Mill, this was shaved down to 16.7 seconds for a 25 percent gain. The other programs, SpiroDemo and Hilbert, which contained a lot of turtlegraphics, ran 7 percent and 9 percent faster, respectively.

Finally, we decided to test a program of our own which would calculate the squares of the numbers 1 to 10,000. No I/O was performed, so this would be a good test of the *Speed-Up Kit*'s number crunching ability. Results of the test were astonishing. What took 47.8 seconds without the Mill now took 54 seconds—that is, 13 percent slower. After double-checking the results, we reconfigured the system to give an indication of which processor was being used. When you install the new System.Apple, you can request continuous display of "6502" or "6809" in the upper-right corner of the screen according to which processor is active at the moment. Although the Mill's 6809 can operate simultaneously with the Apple's 6502, it appears that the *Speed-Up Kit* does not take advantage of this feature.

When we re-ran the 10,000 squares program, it became clear that the program was spending almost 50 percent of its time with each processor. This would seem to indicate that the number crunching routines are handled by the 6502 instead of the 6809. In any case, when you add the overhead necessary to switch between the two processors, the overall time required for running this program increases. By the way, when the program was timed with the running CPU indicator, it ran an additional 4 percent slower as a result of having to update the screen every time the program switched processors.

Stellation II has announced an update that will enable the 6809 to do floating-point math. The update—which was not available for review—sells for $45.

A Gold-Plated Toothpick? At $369 for the complete *Speed-Up Kit* including the Mill, this purchase may be a hard one to justify for the modest speed improvements gained. However, most of this cost is for the hardware, which can be used for other applications. If you can envision the future power of your Apple with the Mill, you may wish to enjoy the benefits of the *Speed-Up Kit* in your Pascal work now while waiting for promised developments such a 6809/Apple operating system. Better yet, if you are a real software hacker, why not help to modify

3096800068096800068096800068096800068096800068096800

or develop 6809 software for use with the Mill? *The Assembler Development Kit* will make this quite painless.

6809 Assembler Development Kit. A 6809 assembler to run with the Mill, produced by Conejo Computer Products, is available from Stellation Two for $80. Like any good assembler, it makes two passes of the source code, which can be entered in free form. The actual source must be written by a text editor (not supplied) that creates unnumbered ASCII text files such as the one included in Apple's *DOS Tool Kit*.

Upon assembly, a formatted listing will be generated and the object code is written to the disk. Object code is stored as a text file on the disk, making small changes easy to perform with a text editor. Both listing and object code generation can be suppressed either at assembly time or via pseudo-ops in the source code. One reason for suppressing both would be to make a fast check of the source code for syntax errors. Once the object code file has been created, it can be loaded into memory with the supplied loader program.

One of the most interesting features of this assembler is that it uses a combination of 6809 programming, 6502 machine code, and a short Applesoft program all running simultaneously. The assembler itself is written in 6809 code; it is about 8K long and is position-independent (one of the nice features of most 6809 code). The other two portions of the assembler triad allow it to be run from DOS 3.3. Some of the assembler's other features include: compatibility with 6800/6801 syntax, input file chaining for creating libraries; paged, numbered, and titled listings, and direct page setting.

Anyone who has never written 6809 code before is in for quite a surprise. The features of the 6809 are considerably more powerful than those of most other eight-bit microprocessors. Compared to the 6502, it's like night and day. Those of us who have labored over limitations of the 6502 such as restricted stack operations, incomplete addressing modes, and limited branching, for example, will be overwhelmed by the rich instruction set of the 6809. This, combined with the hardware improvements—including sixteen-bit registers, two stack pointers, and a variable direct page—may spoil you.

The 6502 was a great CPU and, because of Apples and other computers that use it well, it will be around for quite some time. But, eventually, for the Apple II to remain a competitive computer, its power will have to expand. Probably the greatest contribution to this expansion will come from the addition of one or more coprocessors.

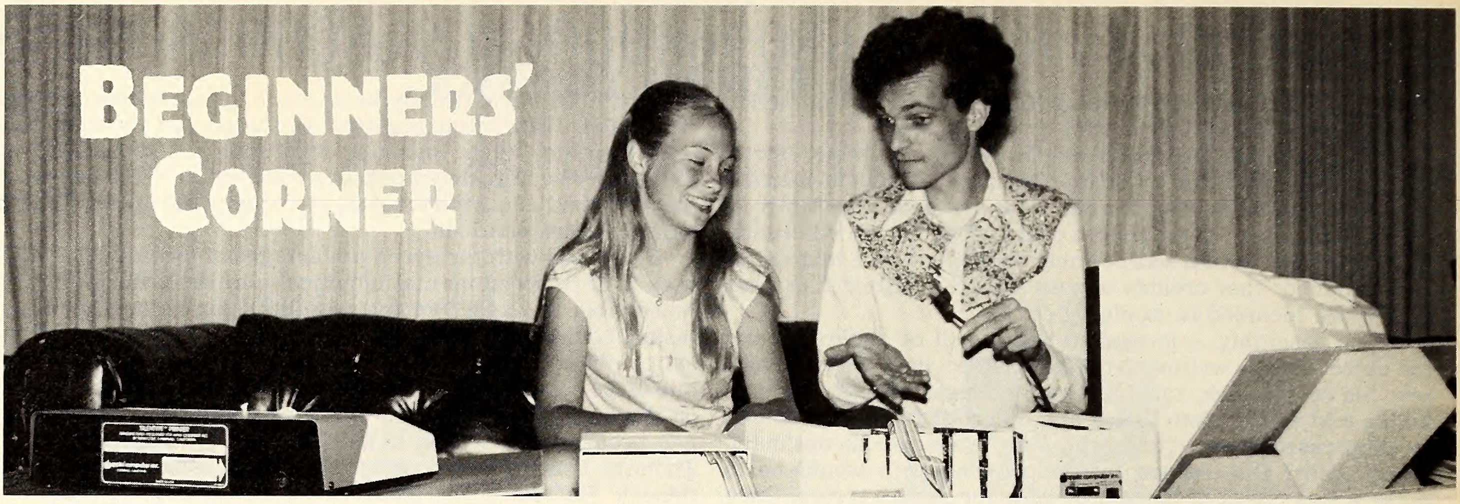
Dtack Grounded and the 68000 Dream Machine. The furor began in late July when the first issue of the *Dtack Grounded Newsletter* surfaced. Where the Mill had promised to double or triple speed improvement, the 68000 attached processor was touted to increase operation speed by ten, twenty or even one hundred times. Three newsletters containing a wealth of information to support these claims have since been published. Because of the outstanding potential importance of this addition to the Apple, we'll be following it closely and will try to keep you up on its progress.

We'll begin by taking a hard look at the situation as it presently stands. First off, it must be noted that this system was first designed to work with a Commodore Pet computer. By early November, a working prototype of the Apple board should exist. The complete system will consist of an Apple interface card, the 68000 board, which will reside outside the Apple and, if needed, a separate power supply. The 68000 board will contain up to 92K of its own RAM with additional 128K memory expansion boards in the wings. Considering the 68000's total address space of 16.7 megabytes, you can see where this is leading. Pricing for a starter system (4K) will run as low as $650.

As in the case of the Mill, the great potential of this device lies in the software. The primary interest of Dtack Grounded right now is to tie their hardware into the floating point math routines of Basic interpreters such as Applesoft. The necessary patches have already been made to the Microsoft Basic on the Pet. From all indications, replacing just the four basic math routines (add, subtract, multiply, and divide) in the interpreter so that they are performed by the 68000 instead of the 6502 results in a thirteenfold increase in the speed of mathematical computations.

Combined with the speed advantages of a compiler (which does little to speed up mathematical number crunching), it is reasonable to expect an average program to run ten times faster. Looking down the road a bit further, it seems only logical to place the entire combining program (either compiled into 68000 code or with a 68000 interpreter) and its data storage on the 68000 board with the Apple becoming little more than I/O processor, handling keyboard, display, disk, and so on. Then the floating point math routines will again become a bottleneck. But, for a few bucks more, a new floating point processor chip, the 8087, promises to solve this problem. Putting all this together—a 68000 CPU board running at 8MHz, the 8087 floating-point hardware and a 68000 compiled Basic—we are talking about a computer on which programs will run one hundred times faster than they do currently in Applesoft. The total added cost of such a system may be as little as $3000. But don't reach for your checkbook yet. Most of the above is at least six months away.

If all of the above has your head spinning, it's no wonder. While some people are turning to newer computers from Xerox or IBM to gain more power than a stock Apple, we are fortunate enough to have a few entrepreneurs who would rather increase the power of the already popular Apple II. The Apple has a broad user base, a wealth of software, and a five-year lead on the competition. With powerful new add-ons like the Dtack Grounded 68000 board, there can be little doubt that the Apple II will be around for quite a while.



BY CRAIG STINSON

Everything the computer does involves numbers in some fashion. Even when the Apple is moving words around on the screen, it's still manipulating numbers behind the scenes. And even when you address the computer in a high-level language like Applesoft, you are still communicating with it numerically. So before we get to the subjects of programming and languages—as we will in due time—we need an introductory discussion of numbers and how they are handled within the computer.

Thank Goodness We Weren't Designed To Have Fifty Fingers. Try to imagine how high you could count if every quantity had a unique numeric symbol, completely unlike the symbols for every other quantity. Or, looking at the same issue from another angle, suppose the English language had entirely unrelated words for every imaginable number. How high would we bother to count?

The difficulty of counting under such conditions could be attributed to more than just the need to memorize a lot of different symbols. That in itself is not an overwhelming job for the human mind, as the existence of pictographic languages demonstrates.

Without some kind of numbering system that not only denotes quantities but also indicates relationships between them, all amounts of appreciable size might well seem alike to our minds. And computation would be a formidable job.

Fortunately, somewhere in antiquity some very bright people devised a method of ciphering that uses only a handful—or to be exact, a double handful—of symbols. And thanks to their genius, we can represent any rational number we like with the ten digits we learned as children.

The system is called a *radix,* or positional system. In this kind of system, each figure in a number represents the product of its digit value times a value determined by its position within the number. For example, in the number 456.7, the 4 really means four times one hundred; the 5 stands for five times ten; the 6 means six times one; and the 7 represents seven times one-tenth. The period in this number is called the radix point.

To generalize about positional numbering, the position directly left of the radix point represents ten raised to the power of zero (which is equal to one), the next position to the left is worth ten to the first power, the next is worth ten to the power of two, and so on for as many positions as we wish to add. Moving to the right of the radix point, the first position represents ten to the power of minus one, the second stands for ten to the power of minus two, and so on.

Not a bad scheme. To get an idea of its power, try performing some calculations using a nonpositional numbering system like that of Roman numerals. Or for that matter, try writing a number like ten million in Roman numerals. You'll need a lot of room, a lot of patience, and a lot of ems.

Both the positional system we use in our everyday lives and the Roman numeral system have origins in anatomy. There are Roman numerals for five, ten, fifty, one hundred, and so on, and the positional system we use has a radix, or base of ten. Both systems reflect the fact that we are pentadactyl creatures; like many other animals, we have ten instruments at the end of our hands with which we manipulate the world. The word *digit* itself reflects this anatomical origin; *digitus* in Latin means pointer and, hence, finger.

But while there may be profound reasons why the animal kingdom has evolved pentadactylly, nothing inherent in logic dictates that we count in fives or tens. And there are good reasons why computing machines generally count in twos.

The principal reason computers use the *binary*, or base two, counting system is that binary digits have only two possible values—zero and one. And there are numerous physical devices that can be made to exist reliably in either of two states as a means of representing zero or one. Transistors, for example, can be made to produce either a zero voltage or a potential of about five volts. There are no shades of gray between these two possible states.

The computer's memory is an enormous aggregation of such devices. Each one holds one binary digit. In computer terminology, the term *binary digit* is usually contracted to the word *bit*.

Counting in the binary system is just like counting in decimal, except for the different base or radix. To evaluate a binary number, like 1011, assign the column at the right a value of two to the power of zero (which equals one; any number raised to the zeroth power is one); the next column to the left gets the value of two to the first power (which equals two); its neighbor on the left is worth two squared (four); and the last column means two to the third power (eight). So the figures 1011 in binary stand for the quantity one times eight plus the quantity one times four plus the quantity zero times two plus the quantity one times one, all of which adds up to thirteen. The quantity thirteen is written 1101 in binary and 13 in decimal.

If 1101 still looks more like eleven hundred and one to you than like thirteen, don't be disappointed in yourself. You've spent a lot of waking and sleeping moments establishing the mental connections between numbers and their decimal representations. Besides, for most of your work with your Apple you won't need to make quick translations between binary and decimal. What's useful is to understand the principle of the binary numbering system and why computers use this system.

What about fractional or negative numbers, you may ask. They, too, can be spelled out in binary figures. The binary number 10.1, for example, represents the value two and one-half. However, the computer does not represent fractional values this way. Remember, there are no punctuation symbols available to the inner Apple—only bits. The computer has other ways of storing fractional and

negative numbers, and we'll deal with these topics momentarily.

Uncloaking the Mysteries 255 and 32767. The 6502, your Apple's brain chip, is what's called an eight-bit microprocessor. That means it always works with data one byte at a time. The chip's architecture is such that it cannot examine more than eight bits during one machine cycle.

If you do a little arithmetic, you'll discover that with eight bits you can make binary numbers representing quantities from zero to two hundred fifty-five. That's not a very large range, and the computer certainly needs to be able to deal with quantities larger than two hundred fifty-five.

Reading two bytes consecutively and treating them as a single sixteen-bit number extends the range of possibilities to 65,535. That's a little more promising.

There's a whole category of numbers that the Apple treats in this fashion, as two-byte quantities. The name for this category is *integer*. In ordinary parlance, an integer is just a whole number, however large. In the context of the Apple's organization, however, integer has a more restricted meaning. An integer quantity is one that can be stored in two bytes.

To accommodate negative numbers, the 6502 was designed to regard half of that sixteen-bit range as positive and the other half as negative. The largest positive integer value then is 32,767, represented in binary as 0111111111111111. The next binary number, 1000000000000000, is interpreted by the 6502 as −32,768, and what would appear to be the largest possible sixteen-bit binary number, 1111111111111111, actually means −1 to the powers that be within your Apple.

Again, what's important to grasp here is the concept. You'll probably never need to write minus sixteen thousand in binary, but you might someday accidentally overflow the Apple's integer capacity and get an error message like *** >32767 ERR. If that should come to pass, you'll at least know what happened (if not immediately *how* it happened) and the reason for that thirty thousand-odd limit.

Integer Basic is a language that allows only integer quantities. If you try to get Integer to print the number forty thousand, you'll get the overflow error message just described. If you ask Integer to print a fractional quantity, you'll get the message *** SYNTAX ERR.

Applesoft, on the other hand, is capable of recognizing and computing with both fractional numbers and with numbers larger than 32,767. Applesoft deals with numbers internally by means of a method called floating-point representation—which is why Applesoft itself is sometimes nicknamed floating-point Basic and why, when you've been working in Integer and want to switch back to Applesoft (assuming you have both languages available), you type FP.

How'd That E Get Into My Number? Before we can say much about floating-point representation, we need to digress briefly to another numeric topic—scientific notation. This is a shorthand method for writing very large or very small numbers. People who deal with quantities like the distance to Andromeda or the wavelength of an x ray get tired of writing out columns of zeros. The shorthand technique is to express the value as a small number (usually between one and ten) multiplied by ten raised to a certain power. For example, the speed of light can be expressed as 3.8×10^{10} (instead of 38,000,000,000) centimeters per second. Or the length of a carbon-carbon bond can be specified as, say, 1.5×10^{-10} meters.

You may occasionally find your computer speaking to you in scientific notation. If, for example, you have Applesoft running and you type PRINT 9300000000 and hit return, your Apple will respond with 9.3E+09. This is its way of saying 9.3×10^9. Its character generator can produce numbers of only one size, so it can't show that little number as a superior figure. Instead it displays an E to indicate that the two numbers following are an exponent; it is understood that the numbers following the E are an exponent of ten, so the E means times ten to the power of. The plus sign is there to indicate that the exponent is positive.

Applesoft will ordinarily report all numbers larger than ten to the ninth or smaller than ten to the power of minus two in this style of scientific notation. You can speak to it in the same fashion if you like, or in ordinary numbers. But if you ask it to print a number outside those specified limits, you will get it in the form described here. For your part, if you want to enter a number in scientific notation at the keyboard, you have the option of omitting the plus sign when the exponent is positive, and you don't need a leading zero when the exponent is less than ten. If you type PRINT 1E9, for example, the computer will understand you and respond with 1E+09.

To translate numbers from scientific notation into ordinary, nonexponential form, here's what you do. If the exponent is positive and the mantissa—that's the quantity to the left of the E—is a whole number, simply write down the mantissa and start writing zeros. The value of the exponent is the number of zeros you need to write. For example, 4E+12 is equivalent to 4 followed by twelve zeros.

If the mantissa has a fractional part, as in the case of, say, 4.21E+15, you replace the first zeros with the fractional part of the mantissa. So 4.21E+15 is the same as 421 followed by thirteen zeros.

The situation is a little different for negative exponents. 1E−03, for exam-

ple, is equivalent to .001. You just put the mantissa in the *n*th column to the right of the decimal point (where *n* is the value of the exponent) and fill the rest out with zeros. If the mantissa has a fractional part, you add that to the right; 3.09E−07 would be translated .000000309.

Floating Back To Applesoft. For quantities within the range of our ordinary experience, it's usually easier to recognize and deal with the ordinary nonexponential notation than with the scientific style. However, any number that can be written in the customary fashion can also be written as a mantissa and exponent if you have a mind to do so, and your Apple—for its internal purposes—is in fact often so disposed.

Back to floating-point notation. The way the Apple often represents to itself a quantity that isn't a whole number or that lies outside the range between plus and minus 32,767 is rather like scientific notation. It breaks up the quantity into a mantissa and an exponent. The computer reserves five bytes for the storage of a single quantity in floating-point representation. Most of the forty bits are taken up by the mantissa, with the remainder given to exponent and sign.

The fact that a floating-point number takes up so much more room in the computer's memory than a two-byte integer means that programs that need to treat numbers in this fashion may run somewhat more slowly than programs that can use integer arithmetic. Applesoft does all its calculating using floating-point numbers—even if the numbers are whole numbers within the range of plus or minus 32,767—and Integer uses only integers. As you might expect, then, a program written in Integer Basic will be executed more quickly than an equivalent program in Applesoft.

That doesn't mean that Integer is preferable to Applesoft in all cases, but rather that within the range of its capability, it's the faster of the two dialects.

Your Apple Puts the Hex on You. There's one more type of numbering scheme that you're likely to encounter in your intercourse with the Apple. That's hexadecimal. The word *hexadecimal* is a concatenation of the Greek for six and the Latin for ten. Sounds ominous already, doesn't it?

The hexadecimal system is a positional system with a radix of sixteen. As with decimal and binary, to determine the value of a number written in hex, you multiply the figure in the rightmost column times one (radix to the power of zero), the figure to its left by sixteen (radix to the first power), and so on, and sum the results. So 10 in hex, for example, stands for the quantity sixteen, not ten; 100 stands for two hundred fifty-six.

Since our familiar Arabic numerals go only as high as nine, and the numerals *10* in hex mean sixteen, some more symbols are required to represent the numbers ten through fifteen. For these values we use the letters A through F. The quantity ten in hex is represented by the hex digit (hexit?) A; fifteen is represented by F; and, to look at a slightly more complicated example, 3D in hex stands for sixty-one (the product of three and sixteen added to the product of thirteen and one).

If you find it strange to see letters used as numerals, think of the set of hex digits as a modified and extended suit of cards. In the context of a card game, you're probably accustomed to recognizing J, Q, and K as equivalent to eleven, twelve, and thirteen (unless, of course, you play only blackjack or pinochle); hex values will become automatic, too.

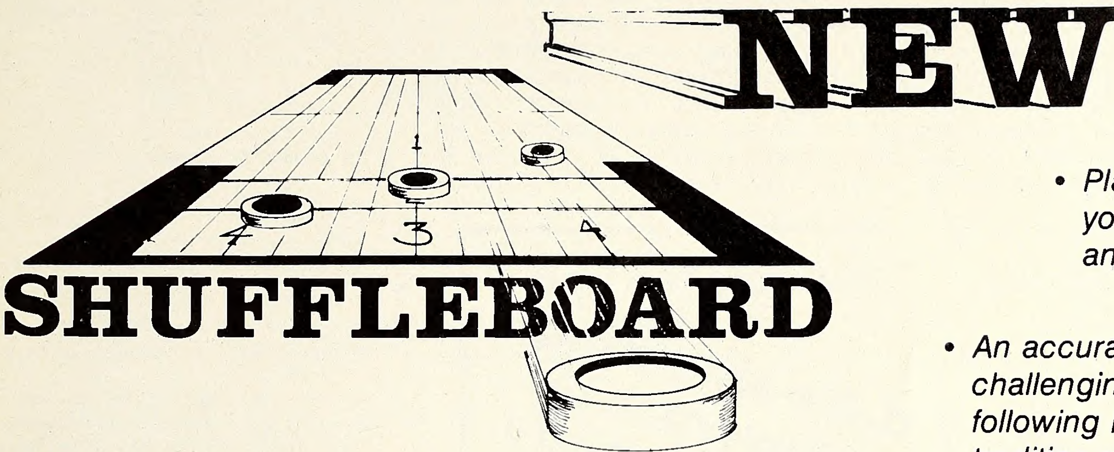
The hex system is used primarily for the convenience of people who program in machine language. The 6502, on a fundamental level, works only in binary. But since it's an eight-bit machine, it always looks at data in eight-bit units. People who deal with the machine in its native tongue must, therefore, also work in multiples of eight bits.

Nibbling at Base Sixteen. For human beings, the writing and reading of numbers like 11000110 is both tedious and hazardous. It's easy for us to make mistakes looking at numbers like that—probably because our sensory systems are accustomed to more variety and contrast than is afforded by a long binary number. The hexadecimal system is an elegant way to simplify the job.

Four bits in a binary number can cover all the ground between zero and fifteen. So can a single hex digit. Eight bits—one byte—of binary code can, therefore, be represented by two hex digits standing side by side. Go back to that binary number in the previous paragraph as an example. To convert it from binary to hex, first split the eight bits into four-bit segments, then evaluate each four-bit unit—called a *nibble,* believe it or not—as though it were a binary number all by itself. The four bits on the left, 1100, are equivalent to eleven, which in hex is B. The four bits on the right, 0110, stand for the value six, which in hex is 6. So the binary number 11000110 can be written in hexadecimal as B6. Once you get used to it, you'll find that converting eight or more bits to hexadecimal is simpler and more direct than converting them to decimal values.

If you decide to take up machine language programming, you'll find yourself working almost exclusively in hex. Even if you don't become a machine language programmer, it's very likely you'll have occasion to use the system Monitor from time to time. And the Monitor likes to be spoken to hexadecimally. The internal layout of the Apple's memory, moreover, is best understood and almost always discussed through the medium of hexadecimal numbers. That subject — the microanatomy of the Apple's RAM— will occupy us in the next installment of this column.

Innovative Design Software, Inc.
ANNOUNCES
SHUFFLEBOARD
for your APPLE II.™
only $29.95

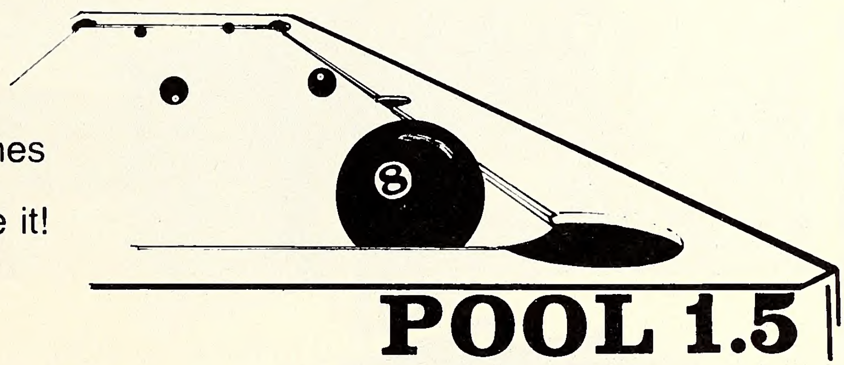


- *Real time HIRES Color Graphics*
- *Play Against your APPLE or another opponent*
- *An accurate and challenging game following in the tradition of Pool 1.5*

— Another first from IDSI

POOL 1.5 features

- Realistic, life-like motion
- HIRES Color Graphics
- Choice of 4 popular pool Games
- You've Got to see it to believe it!
- Only $34.95



"IT (Pool 1.5) is so realistic, you begin filling in the details of the pool hall yourself"
— Softalk Magazine

Both of these excellent games require a 48K APPLE II/Plus and a DISK II. Pool 1.5 requires a set of game paddles or Joystick. Order directly from IDSI or see your nearest APPLE dealer.

IDSI
P.O. BOX 1658
Las Cruces N.M. 88004
(505) 522-7373

- Apple II/Plus is a Trademark of Apple Computer Inc. Pool 1.5 is a trademark of IDSI

MasterCard OR VISA
We accept Visa, MasterCard, Check or Money Order.

THE PASCAL PATH

from page 38

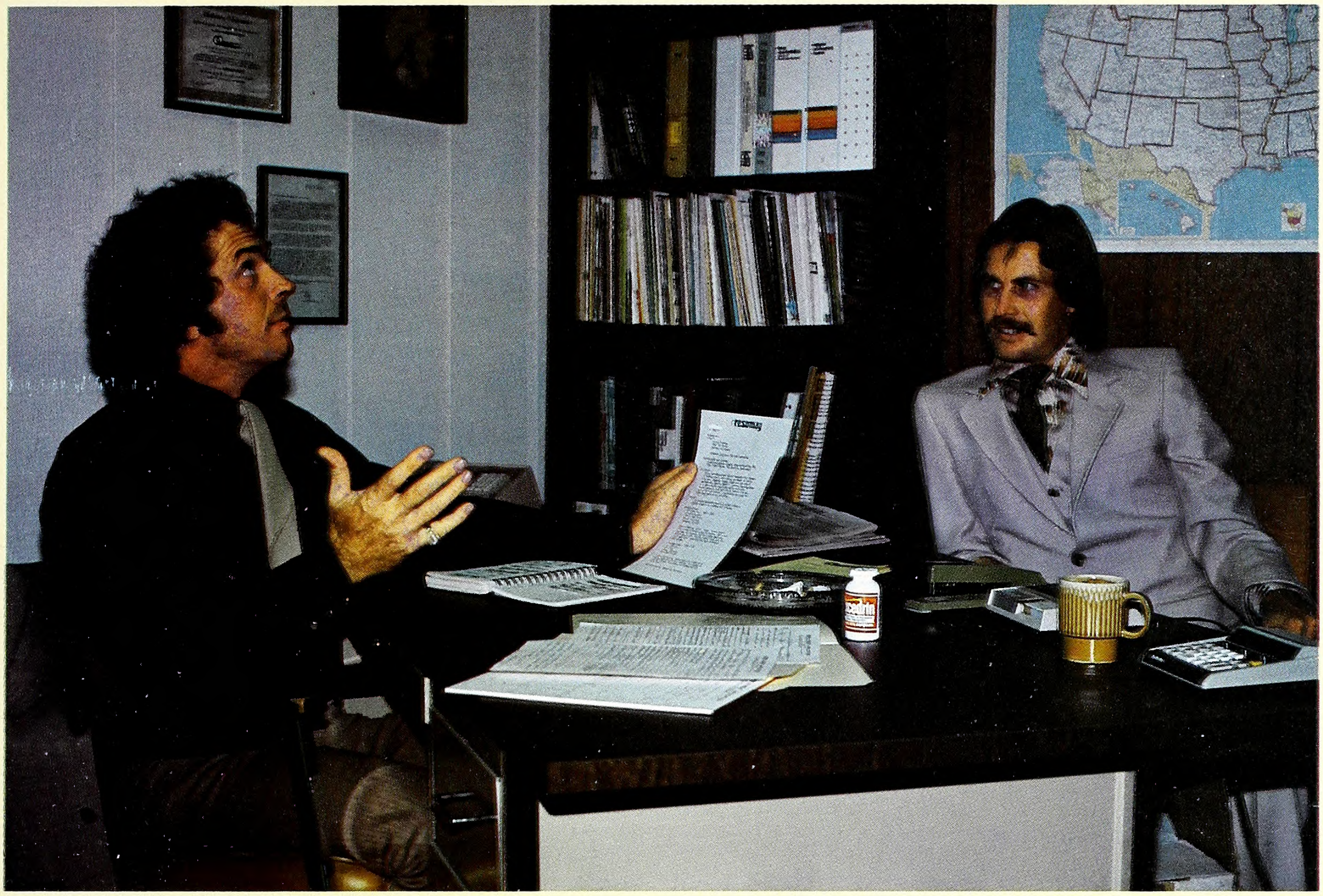
how do you know what you'll forget, and what you won't, about any particular program? For now, I suppose you should insert comments only when it feels right to you. However, no law says that a program must be commented only while it's being written. If you're looking at a finished program some day, and some uncommented portion confuses you, be sure to insert a comment explaining what's happening, as soon as you decide what it is.

Tying It All Together. Assuming that you're the type who reads along politely during my lectures, but really follows this column to see programs, programs, and more programs, here's one that not only computes and displays compound interest on an arbitrary principal, but also illustrates the use of comments in several contexts and employs WriteLn's output formatting features as well. It's offered without explanation as a test of my abilities to use descriptive identifiers and appropriate comments to create a readable program. If part or all of it remains unclear to you, even after careful reading, let me know.

```
PROGRAM
 Compound;
(* Computes and displays daily compounded
      InterestPaid,
    given constants
      Principal,
      TermOfDeposit,
      PeriodsPerYear, and
      AnnualRate (of interest)
*)

(* Program history:
    Author: Jim Merritt
    Last Modified: 20 SEP 1981, by Jim Merritt (V 1.0)
*)

 CONST
    Principal=1000.0; (* Real, because it's in Dollars
      /Cents *)
    TermOfDeposit= 9; (* Whole years only *)
    PeriodsPerYear= 365; (* Daily compounding *)
    AnnualRate= 19.0; (* Expressed as "Percentage"
      *)

 VAR
    InterestPaid,
    AdjAnnRate, (* AnnualRate as decimal fraction *)
    PeriodicRate, (* Interest accrued each PERIOD *)
    Multiplier, (* current principal * Multiplier =
      principal + interest for one period *)
    New Amount
     :Real;

    PeriodNow
     :Integer;

BEGIN (* Compound *)
 (* Drive periodic Multiplier *)
    AdjAnnRate := (AnnualRate / 100.0);
    PeriodicRate := (AdjAnnRate / PeriodsPerYear);
    Multiplier := 1.0 + PeriodicRate;
 (* Compute total future value of Principal *)
    NewAmount := Principal;

    (* Keep applying Multiplier to growing sum of
       principal + interest,
       until term expires. *)
      FOR PeriodNow := 1 TO (PeriodsPerYear *
       TermOfDeposit) DO
       NewAmount := (NewAmount * Multiplier);

 (* Derive InterestPaid from future principal *)
    InterestPaid := (NewAmount - Principal);

 (* Display results *)
    WriteLn( 'Principal: $',Principal:8:2);
    WriteLn( 'Annual Rate: ',AnnualRate:8:2,'%');
    WriteLn( 'Term (Years): ',TermOfDeposit:8);
      (* Integer value *)
    WriteLn( 'Periods/Year: ',PeriodsPerYear:8);
      (* Integer value *)
    WriteLn( 'Interest Paid: $',InterestPaid:8:2);
END      (* Compound *).
```

By modifying some of the CONST definitions, you can change Compound to work with different principals, interest rates, and so on. Try several runs, using different CONST values for each. When you're finished, if you're not too winded, devise your own programs to exercise the elements of Pascal that we have discussed here. The more programs you write, the more you'll be able to write!

He might have had this job. If only he'd used his DICTIONARY.



One small mistake can make all the difference when you're off to land that big job, sign that new account, or write that report. A misspelled word could be that "one thing" that loses you the big job, a promotion, or simply makes you feel like a fool.

If you use an APPLE II or II PLUS in your writing, you can prevent spelling errors. The **DICTIONARY** works in conjunction with your word processor, data base manager or other report generating software* to find and flag possibly misspelled words in your documents. Over 25,000 words are supported by the **DICTIONARY**, with over a thousand more words for you to define for your personal business usage (for instance, words like "byte", "defacto", and "ROM"). With the **DICTIONARY** and a little proofreading, there is no need to ever worry about misspelled words again (and isn't it nice to have one less thing to worry about when you hand in that report?).

*The **DICTIONARY** reads any standard DOS text or binary file in 3.3. $99.95

The **DICTIONARY**
Because a misspelled word can be more than an embarrassment.

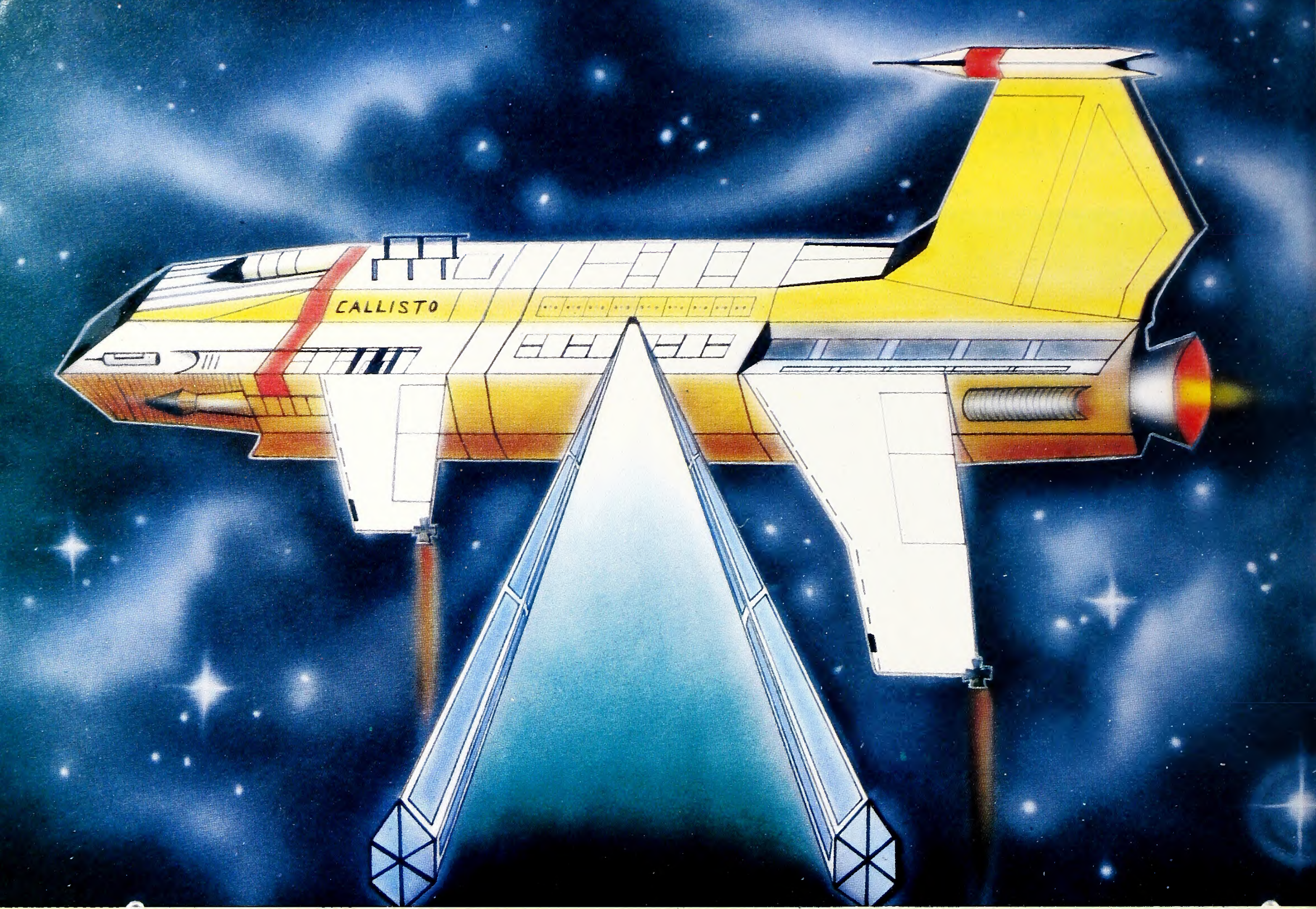
ON-LINE systems

36575 Mudge Ranch Road • Coarsegold, CA 93614 • 209-683-6858

Add $1.00 for shipping

VISA, Mastercard, C.O.D. or checks accepted

DISCOVER THE UNIVERSE



PROUDLY ANNOUNCING

SPACE ADVENTURE

A REAL-TIME SPACE FLIGHT SIMULATOR

Welcome aboard the Callisto.

The Callisto offers you an excellent view of the vast universe before you.

You take the helm and enter the coordinates: quadrant 56, 58, 53; sector 14, 62, 12. Instantly the autopilot takes control of the ship. The viewport reappears as your course automatically changes and you begin your hyperwarp journey. Upon arrival, the message indicator flashes. You take the message and ask the computer for a tactical plot of the situation.

SPACE ADVENTURE is this and much more!

SPACE ADVENTURE offers you real-time instruments, real-time graphics, and an onboard computer that will amaze you.

SPACE ADVENTURE gives you the chance to pilot a space craft through true 3-D space for an adventure you will never forget. **SPACE ADVENTURE** . . . for any Apple II or II+ with 48K and disk. Suggested retail price is $29.95.

Sierra Software

EPISODE TWO COMING SOON
DEALER INQUIRIES INVITED

SIERRA SOFTWARE • 536 E. SAHARA AVENUE • LAS VEGAS, NEVADA 89104

Softalk Presents The Bestsellers

Never before, and probably never again, will any program dominate any bestseller list the way *VisiCalc*, Personal Software's electronic calculator, clobbered the opposition in the month of September.

The potency of the software package has been acclaimed in nationwide publications, and its dominance of *Softalk*'s bestseller list has been well chronicled.

But never has any program had a better than three-to-one margin over its nearest competitor, which is the margin by which *VisiCalc* topped all Apple software packages in September. It appears that the last of the pent-up demand for the program in its 3.3 operating system form was felt at the end of summer.

VisiCalc recorded numbers never before measured by the *Softalk* poll in stretching its lead to previously unimagined lengths.

The width of the margin indicates the vagaries inherent in attempting to apply *Softalk*'s index numbers to a month-by-month analysis. *Raster Blaster,* moving up a couple of notches to second place, was able to record only an index number of 29.50. That contrasts with its index rating of 54.46 last month; but the statistician who then believes that he perceives a 50 percent decline in *Raster Blaster* sales would be sadly mistaken. In fact, the program remained relatively steady in the marketplace, overtaking *Hi-Res Adventure #3: Cranston Manor* and *Gorgon* by virtue of not declining in sales.

Surprises in the Top Thirty were myriad, and the low index numbers, vis a vis *VisiCalc,* should not detract from the outstanding performance improvements of some software packages.

Strategy 5

This Month	Last Month	
1.	—	**Castle Wolfenstein**, Silas Warner, Muse
2.	2.	**Flight Simulator**, Bruce Artwick, SubLogic
3.	1.	**Robot War**, Silas Warner, Muse
4.	4.	**Sargon II**, Dan and Kathe Spracklen, Hayden
5.	—	**Prisoner**, David Mullich, Edu-Ware Services

Adventure 5

This Month	Last Month	
1.	1.	**Hi-Res Adventure #3: Cranston Manor**, Harold DeWitz and Ken Williams, On-Line Systems
2.	5.	**Softporn**, Chuck Benton/Blue Sky Software, On-Line Systems
3.	3.	**Oo Topos**, Michael Berlyn, Sentient Software
4.	4.	**Zork**, Mark S. Blank, Timothy Anderson, Bruce Daniels, P. D. Leblins, Scott Cutler, and Joel Berez/Infocom, Personal Software
5.	2.	**Hi-Res Adventure #2, Wizard and the Princess**, Roberta and Ken Williams, On-Line Systems

Fantasy 5

This Month	Last Month	
1.	1.	**Ultima**, Lord British, California Pacific
2.	2.	**Caves of Karkhan**, Rodney Nelsen, Level-10
3.	—	**Wizardry**, Andrew Greenberg and Robert Woodhead, Sirtech
4.	3.	**Dragon Fire**, Rodney Nelson, Level-10
5.	4.	**Dragon's Eye**, Robert Leyland, Automated Simulations

Business 10

This Month	Last Month	
1.	1.	**VisiCalc**, Software Arts/Dan Bricklin and Robert Frankston, Personal Software
2.	2.	**DB Master**, Alpine Software/Stanley Crane and Jerry Macon; and Barney Stone, Stoneware
3.	5.	**Personal Filing System**, John Page, Software Publishing Corporation
4.	3.	**VisiTrend/VisiPlot**, Micro Finance Systems/Mitch Kapor, Personal Software
5.	—	**BPI General Ledger**, John Moss and Ken Debower, Apple Computer
6.	4.	**VisiDex**, Peter Jennings, Personal Software
7.	6.	**PFS: Report**, John Page, Software Publishing Corporation
8.	7.	**Apple Plot**, Apple Computer
9.	—	**BPI Accounts Receivable**, John Moss and Ken Debower, Apple Computer
10.	—	**Data Factory**, Bill Passauer, Micro Lab


Bill Budge's
Raster Blaster
VIDEO PINBALL FOR THE APPLE II; REQUIRES A 48K APPLE II

The most significant leap was made by Software Publishing's *Personal Filing System,* which rose to fifth from nineteenth. It's the first time in the top ten for *PFS*.

Also coming a long way was *Epoch* from Sirius Software, which moved from twenty-ninth to tenth.

The Top Thirty saw eight new entries this month, three packages rejoining the list after various length absences, and five programs seeing Top Thirty status for the first time. First-timers were *Pegasus II* from On-Line Systems, *Home Money Minder* from Continental Software, *Castle Wolfenstein* from Muse, *Magic Window* from Artsci, and *DOS Boss* from Beagle Bros.

Home 10

This Month	Last Month	
1.	3.	**Home Money Minder**, Bob Schoenburg and Steve Pollack, Continental Software
2.	1.	**Typing Tutor**, Image Producers, Microsoft
3.	2.	**Graphtrix**, Steve Boker, Data Transforms
4.	—	**Goodspell**, Henry G. Baker, Special Delivery Software, Apple Computer
5.	—	**The World's Greatest Blackjack Program**, Warren Irwin, Carl Cooper, and Lance Humble, Special Delivery Software, Apple Computer
6.	4.	**Personal Finance Manager**, Jeffrey Gold, Special Delivery Software, Apple Computer
7.	5.	**ASCII Express**, Bill Blue, Southwestern Data Systems
8.	5.	**VisiTerm**, Tom Keith, Personal Software
9.	—	**Tax Preparer**, Dr. James Howard, Howardsoft
10.	—	**MasterType**, Bruce Zweig, Lightning Software

Hobby 10

This Month	Last Month	
1.	1.	**DOS 3.3**, Apple Computer
2.	4.	**DOS Tool Kit**, Apple Computer
3.	—	**DOS Boss**, Bert Kersey, Beagle Bros.
4.	3.	**Expediter II**, Stewart Einstein and Dennis Goodrow, On-Line Systems
5.	2.	**Complete Graphics System**, Mark Pelczarski, Penguin Software
6.	—	**The Inspector**, Bill Sefton, Omega Microware
7.	5.	**Super Disk Copy III**, Charles Hartley, Sensible Software
8.	—	**Enhanced MX-80 Graphics**, David Hudson, Computer Station
9.	6.	**Multi-Disk Catalog**, Charles Hartley, Sensible Software
10.	7.	**Bill Budge's 3-D Graphics Package**, Bill Budge, California Pacific

Word Processors 5

This Month	Last Month	
1.	1.	**Apple Writer**, Apple Computer
2.	5.	**Easy Writer**, John Draper, Information Unlimited Software
3.	4.	**Magic Window**, Gary Shannon and Bill Depew, Artsci
4.	3.	**Supertext II**, Ed Zaron, Muse
5.	2.	**Word Star**, Micro Pro

Regaining positions in the Top Thirty were *DOS Tool Kit* from Apple Computer, *BPI General Ledger* from Apple Computer, and *Easy Writer* from Information Unlimited.

The Business Ten saw the addition of two accounting packages written by BPI and distributed by Apple. No new programs made the Word Processor Five.

New to the Hobby Ten were *DOS Boss* from Beagle Bros., *Inspector* from Omega Software, and *Enhanced MX-80 Graphics* from Computer Station. Appearing in the Hobby Ten for the first time were *The World's Greatest Blackjack Program* and *Goodspell,* both Special Delivery Software items from Apple that are finding acceptance at the retail store level, *MasterType* from Lightning Software and HowardSoft's *Tax Preparer.*

There were few shakeups in the entertainment software categories. *Castle Wolfenstein* took over the lead in the Strategy Five, but no other new programs made that list. Only a shuffling of positions marked the Adventure Five listing.

A program to be marked edged its way into the Fantasy Five, even though it was in the marketplace for only the briefest time during the month of September. The program is *Wizardry,* which collared third in the Fantasy list and figures to be a contender for Top Thirty listing as well as challenging *Ultima* as the top Fantasy game.

One more caution about the low index ratings should be noted: September was not an awful month for software; it was just a superior month for *VisiCalc*. In fact, September sales give portents of the biggest holiday shopping season yet as Apple owners seem prepared to splurge for Christmas.

For those statistically minded folks with nothing better to do, you can program your Apple to generate more usual index ratings for this month's list by asking it to multiply each index rating by a factor of 2.8. That still doesn't mean much, but it'll look more like you're accustomed to seeing the index ratings look—with the exception of *VisiCalc*, of course.

Apple-franchised retail stores representing approximately 7.8 percent of all sales of Apples and Apple-related products volunteered to participate in the poll.

Respondents we contacted early in October to ascertain their sales leaders for the month of September.

The only criterion for inclusion on the list was number of sales made—such other criteria as quality of product, profitability to the computer retailer, and personal preference of the individual respondents were not considered.

Respondents in October represented every geographical area of the continental United States.

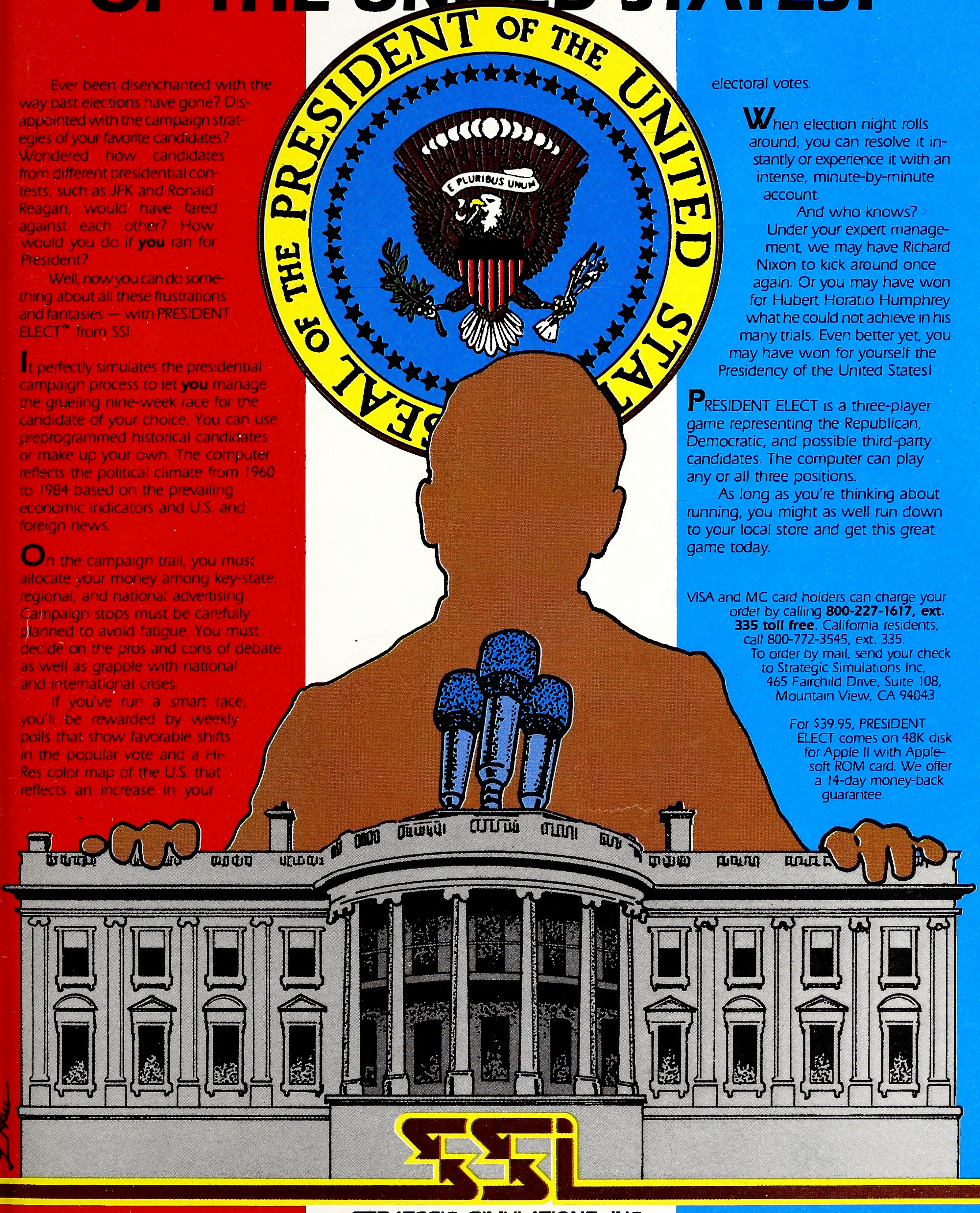
Results of the responses were tabulated using a formula that resulted in the index number to the left of the program name in the Top Thirty listing. The index number is an arbitrary measure of relative strength of the programs listed. Index numbers are correlative only for the month in which they are printed; readers cannot assume that an index rating of 50 in one month represents equivalent sales to an index number of 50 in another month.

Probability of statistical error is plus-or-minus 5.4 percent, which translates roughly into the theoretical possibility of a change of 3.7 points, plus or minus, in any index number.

The Top Thirty

This Month	Last Month	Index Number	
1.	1.	95.60	**VisiCalc**, Software Arts/Dan Bricklin and Robert Frankston, Personal Software
2.	4.	29.50	**Raster Blaster**, Bill Budge, BudgeCo
3.	5.	27.34	**Apple Panic**, Ben Serki, Broderbund Software
4.	6.	26.36	**DB Master**, Alpine Software/Stanley Crane and Jerry Macon; and Barney Stone, Stoneware
5.	19.	24.98	**Personal Filing System**, John Page, Software Publishing Corporation
6.	2.	24.19	**Gorgon**, Nasir, Sirius Software
7.	9.	18.88	**Sneakers**, Mark Turmell, Sirius Software
8.	18.	18.49	**DOS 3.3**, Apple Computer
9.	16.	18.29	**Apple Writer**, Apple Computer
10.	29.	17.11	**Epoch**, Larry Miller, Sirius Software
	3.	17.11	**Hi-Res Adventure #3: Cranston Manor**, Harold DeWitz and Ken Williams, On-Line Systems
12.	7.	14.95	**Snoggle**, Jun Wada, Broderbund Software
	8.	14.95	**Ultima**, Lord British, California Pacific
14.	21.	14.56	**Olympic Decathlon**, Tim Smith, Microsoft
15.	—	14.36	**DOS Tool Kit**, Apple Computer
16.	—	13.57	**Pegasus II**, Olaf Lubeck, On-Line Systems
17.	—	12.79	**Home Money Minder**, Bob Schoenburg and Steve Pollack, Continental Software
18.	—	11.61	**Castle Wolfenstein**, Silas Warner, Muse
	13.	11.61	**VisiTrend/VisiPlot**, Micro Finance Systems/Mitch Kapor, Personal Software
20.	—	11.02	**BPI General Ledger**, John Moss and Ken Debower, Apple Computer
21.	15.	10.62	**Gobbler**, Olaf Lubeck, On-Line Systems
22.	—	10.43	**Easy Writer**, John Draper, Information Unlimited Software
23.	—	10.03	**Magic Window**, Gary Shannon and Bill Depew, Artsci
	11.	10.03	**Pool 1.5**, Don Hoffman, Howard de St. Germaine, and Dave Morock, Innovative Design Software
25.	17.	9.64	**VisiDex**, Peter Jennings, Personal Software
26.	14.	9.05	**Flight Simulator**, Bruce Artwick, SubLogic
27.	10.	8.46	**Robot War**, Silas Warner, Muse
28.	23.	7.87	**Typing Tutor**, Image Producers, Microsoft
29.	24.	7.08	**Graphtrix**, Steve Boker, Data Transforms
30.	—	6.88	**DOS Boss**, Bert Kersey, Beagle Brothers

FILL IN THE NEXT PRESIDENT OF THE UNITED STATES!



Ever been disenchanted with the way past elections have gone? Disappointed with the campaign strategies of your favorite candidates? Wondered how candidates from different presidential contests, such as JFK and Ronald Reagan, would have fared against each other? How would you do if **you** ran for President?

Well, now you can do something about all these frustrations and fantasies — with PRESIDENT ELECT™ from SSI.

It perfectly simulates the presidential campaign process to let **you** manage the grueling nine-week race for the candidate of your choice. You can use preprogrammed historical candidates or make up your own. The computer reflects the political climate from 1960 to 1984 based on the prevailing economic indicators and U.S. and foreign news.

On the campaign trail, you must allocate your money among key-state, regional, and national advertising. Campaign stops must be carefully planned to avoid fatigue. You must decide on the pros and cons of debate as well as grapple with national and international crises.

If you've run a smart race, you'll be rewarded by weekly polls that show favorable shifts in the popular vote and a Hi-Res color map of the U.S. that reflects an increase in your electoral votes.

When election night rolls around, you can resolve it instantly or experience it with an intense, minute-by-minute account.

And who knows? Under your expert management, we may have Richard Nixon to kick around once again. Or you may have won for Hubert Horatio Humphrey what he could not achieve in his many trials. Even better yet, you may have won for yourself the Presidency of the United States!

PRESIDENT ELECT is a three-player game representing the Republican, Democratic, and possible third-party candidates. The computer can play any or all three positions.

As long as you're thinking about running, you might as well run down to your local store and get this great game today.

VISA and MC card holders can charge your order by calling **800-227-1617, ext. 335 toll free**. California residents, call 800-772-3545, ext. 335.

To order by mail, send your check to Strategic Simulations Inc, 465 Fairchild Drive, Suite 108, Mountain View, CA 94043

For $39.95, PRESIDENT ELECT comes on 48K disk for Apple II with Applesoft ROM card. We offer a 14-day money-back guarantee.

As part of our demanding standards of excellence, we use maxell floppy discs.

STRATEGIC SIMULATIONS INC.

Apple is a registered trademark of Apple Computer Inc.